

ООО «Базальт СПО»
Институт Программных Систем РАН

**Девятнадцатая конференция
«Свободное программное обеспечение
в высшей школе»**

Переславль-Залесский, 28–30 июня 2024 года

Сборник материалов конференции
Научное издание

Москва,
МАКС Пресс,
2024

УДК 004.91:378
ББК 32.97:74.48
Д-25



<https://elibrary.ru/zjdppo>

Программный комитет:
А. А. Савченко, — председатель,
А. В. Бондарев, Г. В. Курычий, А. А. Маркина, М. О. Петрова

Д-25 **Девятнадцатая конференция «Свободное программное обеспечение в высшей школе»** : материалы конференции / Переславль-Залесский, 28 июня — 30 июня 2024 г. / отв. ред. Чёрный В. Л. — М.: МАКС Пресс, 2024. — 202 с.
ISBN 978-5-317-07203-2
DOI: <https://doi.org/10.29003/m4152.978-5-317-07203-2>

В книге собраны материалы конференции, одобренные Программным комитетом Девятнадцатой конференции «Свободное программное обеспечение в высшей школе».

Ключевые слова: программное обеспечение, Open Source, свободное программное обеспечение, образование, Linux, цифровизация образования.

УДК 004.91:378
ББК 32.97:74.48

Program Committee:
A. A. Savchenko, — chairman,
A. V. Bondarev, G. V. Kouryachy, A. A. Markina, M. O. Petrova

Nineteenth conference ‘Free Software in Higher Education’ : conference materials / Pereslavl-Zalesky, June 28–30, 2024. / ex. ed. Cherny V. L. — M.: MAKSS Press, 2024. — 202 p.
ISBN 978-5-317-07203-2

DOI: <https://doi.org/10.29003/m4152.978-5-317-07203-2>

The book contains theses of talks approved by the Program Committee of the Nineteenth Conference ‘Free Software in Higher Education’.

Keywords: software, Open source, free software, education, Linux, digitalization of education.

ISBN 978-5-317-07203-2

© Коллектив авторов, 2024
© Оформление ООО «МАКС Пресс», 2024

Программа конференции

28 июня, пятница

12.00-13.30 Заселение, обед, регистрация участников

13:20 Автобус от гостиницы «Переславль»

13:50 Автобус от гостиницы «Переславль»

Дневное заседание 14.30–18.35

14.30 Приветственное слово организаторов

14.40–15.00 А. А. Якушин

Свободные лицензии: бремя прошлого или инструмент
будущего 10

15.00–15.20 Г. В. Курячий

Кризис «средневековой» модели образовательной
площадки в условиях информационной связности 11

15.20–15.40 Н. Н. Непейвода

Ещё раз о проблеме списывания — и роли СПО в этом
вопросе 14

15.40–15.55 И. А. Туманов

Внедрение СПО в инфраструктуру образовательной
организации: проблемы и решения 17

15.55–16.10 А. С. Черепанов

«Альт Платформа» как основа ИТ-инфраструктуры вуза .. 19

16.10–16.30	А. В. Драгунов, К. А. Драгунов, Л. С. Гладченко Создание открытой образовательной среды на базе операционных систем «Альт»	23
16.30–16.50	Кофе-пауза	
16.50–17.10	Н. Е. Шалаев Язык R и высшая математика для студентов-политологов: опыт построения практико-ориентированного курса ...	27
17.10–17.30	Г. В. Курячий Курс «Сети в Linux»: опыт пяти лет преподавания	30
17.30–17.45	А. А. Смирновский Программирование для непрограммистов: какой язык программирования учить специалистам по численному моделированию?	32
17.45–18.05	С. А. Быков, А. С. Малетина, Л. А. Корунева Использование свободного ПО в обучении по направлению «Бизнес-информатика»	37
18.05–18.20	А. Г. Михеев Курсовые работы и производственная практика, связанные с проектами СПО, в Финансовом университете и НИУ ВШЭ	40
18.20–18.35	М. Р. Полякова Образовательная среда «Аврора»	45
19:00	Автобус в гостиницу «Переславль»	

29 июня, суббота

9:00 Автобус из гостиницы «Переславль»

9:30 Автобус из гостиницы «Переславль»

Утреннее заседание

10.00–13.00

10.00–10.20	И. А. Хахаев Позиционирование свободных программ для моделирования в электронике	47
-------------	--	----

10.20–10.40	Е. В. Татьяна	
	Использование свободного программного обеспечения при подготовке будущих учителей информатики в области 3D-печати	51
10.40–11.00	Д. С. Литвинов	
	СПО и проектирование интерфейса пользователя	54
11.00–11.20	М. А. Тергоев, А. А. Давыдов, Г. А. Орлов, Г. В. Янковский	
	Разработка программы с графическим интерфейсом для запуска игр с помощью геймпада	58
11.20–11.40	Кофе-пауза	
11.40–12.00	Е. М. Бесчервтная, А. О. Волкова, В. Г. Агабекян	
	Симуляция пилотирования летательного аппарата с помощью информационной системы управления учебным макетом	61
12.00–12.20	М. Р. Коул	
	Доработка СПО по запросам образовательных учреждений	64
12.20–12.40	А. В. Пименов	
	Открытая библиотека методов анализа и генерации векторной графики	67
12.40–13.00	С. А. Фомин	
	PyAlgovizualizer — эффективное преподавание алгоритмов .	69
13.00	Автобус в гостиницу «Переславль»	
13.00–15.00	Перерыв на обед	
14.40	Автобус от гостиницы «Переславль»	

29 июня, суббота**Вечернее заседание
15.00–18.35**

15.00–15.20	С. В. Фомченков, О. И Щавелев	
	Создание эффективных рабочих пространств в мире свободного программного обеспечения	75
15.20–15.40	М. И. Гущин	
	Проект открытого кода научных исследований ФКН	78
15.40–15.55	Ю. А. Андреев, С. А. Симонов	
	Свободное Программное Обеспечение в Передовой Инженерной Школе	82
15.55–16.10	А. Н. Старинин	
	Использование сервисов хранения репозиторий git в обучении студентов	86
16.10–16.25	А. Н. Гетманов	
	Создание научного open source-сообщества: опыт ИТМО...	90
16.25–16.40	А. Г. Уймин, И. М. Морозов	
	Аспекты применения СПО в рамках разработки заданий ДЭ по специальности 09.02.06 Сетевое и системное администрирование	92
16.40–17.00	Кофе-пауза	
17.00–17.20	М. О. Петрова	
	Как помочь преподавателям построить учебный процесс на основе свободного ПО	98
17.20–17.40	П. П. Чайкин	
	О необходимости разработки открытого стандарта маркировки контента, генерируемого нейросетями	99
17.40–17.55	Д. А. Астраханцев, Г. В. Курячий, Д. Ю. Волканов	
	Интерпретатор частично рекурсивных функций	107
17.55–18.15	А. Н. Непейвода, А. Д. Дельман, Д. П. Князихин и др.	
	Chipollino: как лабораторная работа превратилась в инструмент для исследований	111

18.15–18.35	А. В. Речицкий, А. В. Брагин	
	ReactOS — Виндоуз почти здорового человека без ГМО и слежки	115
19:30	Автобус в гостиницу «Переславль»	

30 июня, воскресенье

Утреннее заседание

10.00–13.30

09:00	Автобус от гостиницы «Переславль»	
09:30	Автобус от гостиницы «Переславль»	
10.00–10.20	П. Б. Жданович, О. А. Петрухина	
	Применение СПО для построения виртуальной образовательной ИТ-среды на базе персональных компьютеров	116
10.20–10.40	В. В. Павловский, И. М. Морозов	
	Автоматизация киберполигона в образовательном процессе высших учебных заведений	119
10.40–11.00	А. И. Захарова, Е. А. Погорелко	
	Разработка средства управления пулами виртуальных машин для учебных заведений	124
11.00–11.20	А. А. Сурганов	
	ВиЛКа — Виртуальная Лаборатория Кактус	127
11.20–11.40	П. А. Леляев	
	Система управления компьютерным классом для МОС	129
11.40–12.00	Кофе-пауза	
12.00–12.20	С. С. Быковский, И. В. Степанова, Р. Ф. Халабия	
	Опыт применения свободного программного обеспечения для организации серверов удалённой загрузки в образовательной организации	133
12.20–12.40	П. А. Макаров	
	Практика применения свободного программного обеспечения в преподавании дисциплин физико-математической и технической направленности	136

12.40–13.00	А. М. Васильев	
	Переход компьютерного класса 7-го учебного корпуса ЯрГУ на ОС Simply Linux	141
13.00–13.20	Е. Н. Прокофьева, Л. Б. Чашкин, К. Е. Букин	
	Alt Linux как среда решения образовательных, инженерных и научных задач	144
13.20–13.40	Л. Б. Чашкин, Е. Н. Маршутина	
	Расширение возможностей проекта VPCS при моделировании IPv6-адресации с использованием программного эмулятора компьютерных сетей GNS ₃ ..	151
13.40–14.00	М. А. Черноморец	
	Использование инструмента ipcalc в рамках демонстрационного экзамена по специальности 09.02.06 «Сетевое и системное администрирование» в колледже	155
14:30	Автобус в Москву	

Вне программы

А. Б. Вахранев

	Свободное программное обеспечение в Волгоградском филиале РГУ СоцТех или «История одной аудитории»	159
Д. Костюк, А. А. Маркина, А. Шульган		
	Адаптация университетской программы по алгоритмизации и программированию на языке Python для дополнительного довузовского образования	161
Р. С. Крашенинников, И. И. Митричев		
	Разработка библиотеки Python для решения задач фармакокинетики	166
С. А. Мартишин, М. В. Храпченко		
	Представление конфиденциальных данных в агрегированном виде с использованием векторного формата SVG, языков R и Rust	170
С. А. Мартишин, М. В. Храпченко		
	Распараллеливание поисковых запросов к веб-серверу с использованием языка Rust	174

Г. Г. Гриценко, В. П. Фраленко

Использование открытого программного обеспечения в
задачах обнаружения объектов в видеопотоке 177

А. А. Нозик

Научная экосистема KScience 186

В. П. Андреев

Использование свободного ПО в учебном процессе в
специальности «DevOps-инженер» 189

О. О. Неманов

Наш опыт использования СПО на уроках информатики ... 193

С. А. Скрипов

К вопросу об использовании свободного программного
обеспечения для управления виртуальными машинами
в учебном процессе 194

А. А. Галактионов, Л. М. Багнюк, Е. А. Погорелко

Подходы к созданию комплекса программных средств
автоматизированного проведения и проверки учебных
лабораторных работ 197

А. А. Зубарев

Использование систем поиска уязвимостей локальных и
сетевых систем 198

Анатолий ДОС Якушин
Москва

Свободные лицензии: бремя прошлого или инструмент будущего

Аннотация

Изменение основных принципов дистрибуции программного обеспечения, массовое использование облачных вычислений, распространение программ, созданных с использованием искусственного интеллекта требует переосмысления использования свободных лицензий и понимания их места в современном мире информационных технологий и перспектив дальнейшего развития СПО.

Ключевые слова: свободные лицензии, копилефт, лицензирование программного обеспечения.

Основные принципы лицензирования программного обеспечения, которым следуют и свободные лицензии, закладывались около полувека назад. Изменение технологий дистрибуции ПО, несомненно, сказывается на базовых подходах к лицензированию, меняет привычные представления о роли и месте лицензий в мире информационных технологий. При этих изменениях свободные лицензии становятся особенно уязвимыми, так как они по своей природе охватывают огромный, заранее не определённый круг разработчиков и пользователей.

Однако практика показывает, что оборотной стороной этой уязвимости является способность оперативно реагировать на происходящие изменения и формировать лицензионную политику, соответствующую текущим задачам. Ярким примером такой реакции является создание и внедрение в практику стандартной общественной лицензии GNU Affero (GNU AGPL). Изучение использования GNU AGPL, как решения проблемы лицензирования ПО для облачных вычислений позволяет выявить механизмы адаптации свободных лицензий при серьёзных изменениях дистрибуции свободного ПО. Сегодня крайне актуальным является решение целого клубка лицензионных противоречий, возникающих при создании, внедрении и использовании систем искусственного интеллекта, в том числе вопросы авторского права на

данные, используемые при разработке ИИ, а также права на результаты деятельности подобных систем. Последние два года OSI пытается терминологически определить, что, собственно представляет собой система ИИ с «открытым исходным кодом», однако до настоящего времени результаты носят весьма условный характер.

Концепция открытого исходного кода была разработана для того, чтобы разработчики могли использовать, изучать, модифицировать и делиться программным обеспечением без ограничений. Но ИИ работает принципиально по-другому, и ключевые концепции не могут быть чётко перенесены из области создания программного обеспечения в область создания ИИ. Сегодня, в принципе понятно, какие компоненты систем ИИ должны быть открыты для изучения и модификаций, чтобы вся система ИИ была признана открытой, однако нет ясности, с помощью каких механизмов возможна реализация этих требований.

Большая работа и напряжённая дискуссия в сообществе по вопросу свободных лицензий в области искусственного интеллекта, значительный интерес к данной проблеме показывает, что сегодня свободное лицензирование остаётся одним из важных факторов создания программного обеспечения и имеет значительный потенциал для дальнейшего развития.

Георгий Курячий

Москва, ВМК МГУ

Проект: UNIX <http://uneex.org>

Кризис «средневековой» модели образовательной площадки в условиях информационной связности

Аннотация

Во время прошедшей пандемии стали очевидны достоинства и недостатки чисто дистанционных методик преподавания. Однако средства поддержки дистанционного образовательного процесса развились настолько, что теперь нет возможности полноценно вернуться к традиционному аудиторному способу: если лекции транслируются общедоступным сервисом, очно на них присутствуют только оператор, лектор и члены фан-клуба лектора. Чем такое положение дел грозит устоявшейся практике университетских занятий и что делать дальше? Ответ на первый вопрос мы знаем по собственному опыту, а вот второй — предлагается к обсуждению на конференции.

Ключевые слова: *дистанционное обучение, онлайн-обучение, аудиторные занятия, практические занятия.*

«Средневековая» модель университетского образования (точнее — образовательной площадки) последние несколько сотен лет устроена так:

- Студенты и лектор набиваются в одно время в душную аудиторию.
- Лектор что-то пишет и рисует на доске.
- Студенты пытаются в меру понимания законспектировать сказанное лектором.
- Когда что-то непонятно — или если лектор сам спровоцировал аудиторию — задают вопросы и получают ответ.
- Если студент пропустил лекцию, он обращается к учебнику или справочнику.
- Практическую составляющую или проработку конкретных материалов делают примерно в тех же условиях на семинарах.

Нельзя сказать, что за эти столетия ситуация изменилась к лучшему. Появились существенно более удобные средства презентации, но их часто используют не по назначению, заменяя *порождение знаний* картинкой с готовым рецептом. Вместо одной дорожущей книги на всех есть целый интернет, но если преподаватель заранее не подготовил ссылок на материалы, с которыми он сам согласен, возможно *недопонимание*. Наконец, современный студент не всегда любит и умеет писать, предпочитая просматривать запись лекции на полуторной скорости.

Помимо этого, устоялась традиция дистанционного обучения — это те же самые лекции, но в записи, плюс подготовка довольно большого объёма план-конспектов, плюс большая степень автоматизации в отчётности (например, автоматическая проверка Д/З), а весь интерактив выносится в дистанционный семинар, сводящийся к ответам на вопросы.

Для своих дистанционных семинаров по Python я «изобрёл» т. н. «доску»: половину экрана на компьютере студента занимает окно удалённого рабочего стола, на котором поочерёдно преподаватель демонстрирует примеры для разбора, а студенты — решения заданий [1].

Наступившая вслед за этим пандемия обнажила и заставила доработать слабые места в такой схеме. Лекции переместились в от-

дельный канал YouTube, в чате которого котором сформировалось довольно своеобразное небольшое сообщество. Для обратной связи вне лекций по любым вопросам использовалась группа в Telegram, а семинары с тем же принципом «доски» переехали в Zoom — не без проблем, разумеется.

Когда пандемия сошла, оказалось, что *большая часть инструментов*, разработанных для дистанта, продолжает эффективно работать. Стрим теперь ведётся прямо из лекционной аудитории, более того, запись эфира обрабатывается и также выкладывается в канал в «цивилизованном» виде. Аудиторный практикум, *ориентированный на полную доступность лекций* (живьём или online), проводится всё с той же «доской» (об этом был отдельный доклад на конференции [2]).

В результате большая часть студентов выбирает «дистанционный» формат лекций: аудиторно, особенно к концу семестра, присутствует не более 10%–15% записавшихся. При этом, что экзаменационные задания они сдают вполне успешно, и в отчётах неплохо видны следы именно лекционной формы подачи знаний. Нельзя исключать и то, что качество лекций ухудшилось, и люди проголосовали ногами — хотя в лицо они говорят обратное: записей + материалов на сайте достаточно.

Увеличивается разрыв между теми, кто эффективно воспользовался дистантом, и теми, кто поддался извечному соблазну «лекция есть в записи, завтра посмотрю... в следующие выходные посмотрю лекции... в конце семестра посмотрю *все* лекции». Вопрос в том, насколько вообще нужно бороться за внимание и мотивацию вот этих последних — ленивых, но вполне типичных студентов?

Главная проблема: очень не хватает обратной связи с аудиторией. Это не только мотивирующий фактор для лектора, но и эффективный инструмент выявления «узких мест» в изложении. Не бывает «глупых вопросов» — бывает материал, не дотягивающий до «умного ответа»! Последующее отложенное общение в Telegram-группе, увы, не решает проблему *своевременности*.

Возможно, имеет смысл обновлять лекционные видеоматериалы раз в несколько лет, а живые занятия устроить совсем по-другому. Условно говоря, разбить их на три части: дополнения/ исправления к лекциям, практика а-ля семинар и собственно обратная связь? А как сделать её массовой, как преодолеть «отложенную актуальность»?

Так что средневековой традиции мы пока придерживаемся, но будущее уже наступило, и оно туманно.

Литература

- [1] *Курячий Г.В.* Дистанционное online-преподавание компьютерно-ориентированных дисциплин // Четырнадцатая конференция «Свободное программное обеспечение в высшей школе». Сборник докладов. Москва, МАКС Пресс, 2019, с. 14
- [2] *Балашов В., Курячий Г.* Как мы добывали огонь. Организация аудиторного практикума по курсу «ЯП Python» на останках дистанционного обучения // Восемнадцатая конференция «Свободное программное обеспечение в высшей школе». Сборник докладов. Москва, МАКС Пресс, 2023, с. 60–62

Николай Непейвода

Переславль-Залесский, ИПС РАН

Ещё раз о проблеме списывания и роли СПО в этом вопросе

Ключевые слова: *качество образования, проблема списывания, госэкзамены.*

Проблема списывания стала в последние годы катастрофической для российского технического высшего образования. Причин здесь несколько.

- По традиции отличного советского технического образования сложные задачи и курсовые выдаются большей частью на базе апробированных технических решений и, соответственно, не меняются десятилетиями. А в условиях российского информационного общества достаточно два года не менять курс, и уже все шпаргалки и решения будут в студенческих базах данных. И останется лишь скопировать.
- Достаточно прохладное отношение большинства преподавателей к своим обязанностям означает, что они стремятся минимизировать усилия на проверку работ. И в итоге тот, кто тупо списал, выдаёт правильный ответ, а тот, кто решал сам, с шероховатостями и даже ошибками, с ответом в другой форме, либо с ходом решения и получившимся алгоритмом нестандартным и требующим усилия для понимания. А зачем тратить силы, если

можно просто сравнить ответ? Но даже собственное решение с ошибками в сто раз ценнее скатанного «правильного». А уж про оригинальное и говорить нечего. И в итоге наглые списыватели могут получить оценку даже выше, чем те, кто делает сам.

- Помогает этому система контроля промежуточных «результатов» и процесса, а не итогового выхода. В частности, премирование преподавателей за средний балл.
- Госэкзамены и «бакалаврские диссертации» стали в ряде случаев простой формальностью.

ИТ-специальности даже в ведущих вузах глубоко поражены этой болезнью. Попытки лечения ужесточением контроля списывания приводят лишь к перегибам типа досмотров с металлоискателями или проверки программ с помощью «Антиплагиата». Предлагаемая система контроля квалификации выпускников основана на тысячелетнем опыте Китая, к которому он возвращается сейчас, и на СПО. Но её предпосылкой являются некоторые административные решения.

- Прекратить контролировать учебный процесс в вузах и вместо этого ввести госприёмку выпускников, включающую государственные испытания (не просто госэкзамены).
- Для ИТ-специальностей срок госиспытаний достаточен от 3 до 7 дней. Они должны включать несколько задач разной трудности на темы, которые данный вуз указал как сильнейшие в своих курсах, и составление и отладку программы, при этом задания должны быть по крайней мере три: два на темы, специфические для данного факультета, и одно на темы, указанные им как качественно освоенные. Обязательным условием является использование лишь СПО.
- Теоретические опросы на сильные и качественно освоенные темы. Достаточно тоже трёх, но ответы в свободном стиле требуются на всё.
- Сочинение в свободном стиле, как выпускник представляет себе дальнейшую профессиональную деятельность.

Поскольку важнее для страны иметь больше сильных специалистов, недопустима единая программа госиспытаний для всех вузов. Госкомиссия должна составлять её на базе самоотчёта вуза с указанием своих сильных, средних и слабых сторон и его учебной программы. Идеально составление окончательной программы в помещении,

изолированном от внешнего мира, за последние сутки перед началом испытаний и раздача её выпускникам в момент их начала.

На основе китайского опыта госэкзаменов на степени идеальной была бы организация испытаний следующим образом.

- Иметь в ведущих городах дворцы знаний, состоящие из одностанных келий, экранированных от внешней информации, с удобствами (включая кушетку с бельём для отдыха).
- Вход в коридор келий через раздевалки, в которых выпускники полностью переодеваются в казённую одежду и оставляют все личные вещи.
- Всё время госиспытания выпускник использует по своему усмотрению. Ему предоставляется трёхразовое качественное питание и электрочайник с заваркой и закусками для использования в любое время. Это заодно позволит уменьшить стресс, вызванный решением сложных проблем в сжатое время и в толпе.
- В келье предоставляется компьютер с защищёнными от изменений программами и базами данных (проверять только память человека в наше время абсурдно), всё на базе СПО. Должны иметься редактор (например, TeX) для подготовки ответа и в необходимом количестве программы символьных вычислений и машинной графики. Весьма желательно иметь также проверенные учебники и справочники. И, конечно, необходимы бумага и письменные принадлежности.
- Когда экзаменующийся подал сигнал о завершении работы, ему копируют и заверяют электронной подписью содержимое изменяемой памяти компьютера и по его желанию фото его материалов на бумаге.

И как завершающий штрих, такие предложения могут понравиться чиновникам, но для их проведения в жизнь необходимо с нуля перестроить организацию Минобрнауки.

Иван Туманов

Санкт-Петербург, ГБУ ДПО «Санкт-Петербургский центр оценки качества образования и информационных технологий» (СПбЦОКОиИТ)

Внедрение СПО в инфраструктуру образовательной организации: проблемы и решения

Аннотация

Свободное (и, во многом, отечественное) программное обеспечение в школах (область взаимодействия) уже не редкость, хотя ожидания больше, чем реальная степень импортозамещения. Про специфику внедрения в системе общего образования и хочется поговорить, основываясь на опыте работы в данном направлении в Санкт-Петербурге.

Ключевые слова: *линукс, кадры, администрирование.*

Опыт внедрения СПО в Санкт-Петербурге

Приказов и сроков по импортозамещению в Санкт-Петербурге нет и не было. Специфика региона заключается, на мой взгляд, в наличии финансовых возможностей, что позволяло в прошлые годы приобретать лицензии на MS Windows и Office, с одной стороны, а с другой — заметное количество достаточно квалифицированных специалистов, которые не только эффективно использовали имеющиеся решения (MS Active Directory, в первую очередь), но и пробовали и внедряли свободные и отечественные решения.

В рамках организации обмена опытом и популяризации свободных и отечественных решений в Санкт-Петербурге с 2008 г. (в 2023 г. состоялась XIV) проводится ежегодная конференция «Проблемы и перспективы внедрения отечественных и свободных программных и технических решений в образовании». Осенью такая встреча.

В 2010 г. в Санкт-Петербурге силами компании «Визардсофт» был запущен Школьный проект LinuxWizard, в рамках которого специалистами компании за символическую сумму в школах реализовывалось внедрение собственного дистрибутива, техническая поддержка и обучение. В 2014 году данное решение было развёрнуто в 100 школах города (из 700), но проект постепенно затих по причине невозможности полноценной кадровой и финансовой поддержки только силами разработчиков.

В 2013 г. решением V конференции СПО была создана рабочая группа, включающая представителей Комитета по образованию Санкт-Петербурга, СПбЦОКОиИТ, ИПАП (Институт прикладной автоматизации и программирования), Университета ИТМО, Российской ассоциации свободного программного обеспечения (РАСПО), образовательных учреждений города. Рабочая группа через пару лет распалась. По рекомендации рабочей группы с 2014 по 2016 год два ОУ — ГБОУ школа №169 с углублённым изучением английского языка Центрального района и ГБОУ школа №567 Петродворцового района — были городскими опытно-экспериментальными площадками по теме использования СПО.

Нашим центром несколько раз в год проводятся вебинары, выездные семинары, митапы, связанные с техническими аспектами организации школьной сети.

На примере активно использующих СПО школ можно сказать следующее. Для полноценной инфраструктуры школы инструментов в мире СПО достаточно. SAMBA — как файловый сервер и контроллер домена MS AD, LibreOffice и P7-офис, как офисное ПО, Proxmox как система виртуализации, система резервирования bareos и т. д. Но, несмотря на всё вышесказанное, сейчас уровень проникновения свободных решений в школы ниже, чем был в середине 2010-х годов.

Проблемы и предложения

Есть следующие предложения для активизации импортозамещения ПО в школах:

- Федеральный уровень. ГИА. Есть движение, ждём результатов.
- Федерально-региональный уровень. Обязательные в регионах к использованию информационные системы нужно делать совместимыми с отечественными ОС. Тут вендор (теоретически), как заинтересованная сторона, может задействовать свой вес и связи (которых всё больше, судя по отчётам о внедрении).
- Нужны ориентиры для школ по выбору ПО. Скорее региональный уровень. Пример Пскова — управление образованием перешло на «Альт», региональные системы перевели на свободное ПО, школам готовы оказывать поддержку именно этой ОС.
- Для массового использования нужны типовые технические решения, упрощающие внедрение и обслуживание. Школы раз-

ные, регионы разные и т. п. Тогда уже не так страшна текучка кадров — можно организовывать обучение именно этим типовым решениям.

- Наличие в каждой школе квалифицированного админа — утопия. Для развёртывания и управления типовыми решениями достаточно одного «надшкольного» администратора для нескольких школ.

Выводы

Массовость и эффективность внедрения свободного и отечественного ПО в системе общего образования даст только унификация технических решений и консолидация кадровых ресурсов для управления такими типовыми решениями.

Это потребует перестройки существующих связей в системе управления образованием — такой «надшкольный» администратор или технический отдел может быть частью в некой организации управления образованием, а вот выведение такой техслужбы на уровень услуги (был и такой опыт) чревато частой сменой подрядчиков из-за госзакупок и «взаимонепониманием» школ и таких компаний по вопросам типизации решений и зон ответственности.

Достижение цели массовости импортозамещения ПО не решится только приказами, оно требует выстраивание системы взаимодействия между регионом — школами и разработчиками отечественного программного обеспечения.

Андрей Черепанов

Москва, ООО «Базальт СПО»

Проект: «Альт Платформа» <https://www.basealt.ru/alt-platform>

«Альт Платформа» как основа ИТ-инфраструктуры вуза

Аннотация

Рассматривается архитектура и вопросы использования «Альт Платформы» для сопровождения ИТ-инфраструктуры вуза как для хозяйственной деятельности, так и для образовательного процесса.

Ключевые слова: *альт платформа, инфраструктура, дистрибутивы.*

Технологии сборки

На протяжении почти четверти века ALT Linux Team поддерживает собственный репозиторий **Sisyphus**. Для сопровождения этого, одного из крупнейших репозиториев в мире, были разработаны и используются множество технологий и инструментов. Для того, чтобы репозиторий из множества взаимозависимых пакетов был целостным и содержал меньше ошибок, команда решала следующие задачи:

1. Воспроизводимая сборка. Для того чтобы пакет однозначно собирался в любое время и в любых условиях, необходимо предоставить все необходимые материалы в изолированной среде. Это значит, что среда должна помимо исходного кода содержать минимальный и достаточный набор зависимостей (в том числе инструментов сборки) и не иметь выход в сеть для исключения рисков недоступности необходимых ресурсов или их возможной подмены. Этот инструмент получил название **Hasher**¹.
2. Для упрощения сборки и ведения совместной с апстримом истории был предложен инструмент **Gear**², который позволяет встроить непосредственно в репозиторий правила сборки и файлы **.spec** для сборки пакетов **RPM** и собирать непосредственно из апстримных тегов, минуя промежуточные стадии создания архивов исходного кода и их хранения (как в **Fedora**, например).
3. Проверка на принятые правила через **sisyphus-check**. Например, запрет на размещение файлов вне принятых каталогов структуры **FHS**.
4. Контроль зависимостей и **ABI**. В **Sisyphus** зависимости пакетов генерируются автоматически на основе опакеченного кода и зависимостей исполняемых файлов. При этом зависимости добавляются не только на библиотеки, но и на упомянутые в коде модули различных скриптовых языков. Зависимости на разделяемые библиотеки не просто выставляются по их именам, как в других дистрибутивах, но и учитывают **ABI** библиотек (**set-версии**).

Помимо программ для сборки пакетов был создан инструментальный сборщик дистрибутивов **mkimage-profiles**³. В настоящий момент

¹<https://altlinux.org/Hasher>

²<https://altlinux.org/Gear>

³<https://altlinux.org/Mkimage-profiles>

на основе этих профилей `Makefile` собираются все дистрибутивы ОС «Альт» и иные образы (например, виртуальных машин).

«Альт Платформа» как продукт

«Альт Платформа» — первый в России технологический комплекс по сборке и распространению программного обеспечения и собственных дистрибутивов. Мы взяли следующие компоненты:

- стабильный репозиторий «Альта», качество которого организационно поддерживается жёстким контролем качества собираемых пакетов;
- дистрибутив `ALT Platform Builder`, который позволяет развернуть зеркало стабильного репозитория «Альта» и обновлять его, а также использовать технологии `ALT Linux Team` для надёжной, безопасной и эффективной сборки как пакетов в единый репозиторий, так и дистрибутивов на его базе (`alterator-mirror`, `gear`, `hasher`, `mkimage-profiles`).

«Альт Платформа» входит в российский реестр программного обеспечения (регистрационный номер ПО: 21540) и может быть использована для следующих задач:

- создавать собственные дистрибутивы;
- вести собственный единый репозиторий из различных компонентов:
 - более 50.000 пакетов, собранных и поддерживаемых в стабильном репозитории;
 - собственное ПО;
 - стороннее ПО;
- обновлять из поддерживаемого репозитория собственную инфраструктуру.

Свой поддерживаемый репозиторий нужен для снижения рисков ухода вендоров, возможности самостоятельно комбинировать существующие решения для собственныхборок ОС (что актуально для больших организаций).

При этом не нужно считать, что «Альт Платформа» — это только сборочница. Это неверно и технологически (в «Альт Платформе» не используется `girar`, который используется для сборки в сам `Sisyphus`

и стабильные репозитории), так и методологически (так как упор сделан не на сборку пакетов, а на использование общего репозитория).

Как продукт «Альт Платформа» доступен свободно для создания собственных дистрибутивов под лицензией GNU General Public License. В случае, если кто-то захочет создавать свой дистрибутив под несвободной лицензией и получить полноценную поддержку по договору, это обсуждается в рамках коммерческой истории.

Использование как основы ИТ-инфраструктуры вуза

Ключевые преимущества использования «Альт Платформы»:

- изучение промышленных практик работы с кодом в `Git` и сборки программного обеспечения;
- возможность создать собственный дистрибутив (как для использования внутри вуза, так и для целей обучения);
- использование единой базы программного обеспечения, включая сборки как собственных, так и сторонних компонентов;
- гарантированная совместимость программных продуктов с операционными системами семейства «Альт» при использовании инструментария «Альт Платформы»;
- централизованное обновление ОС «Альт» и собственных дистрибутивов в рамках вуза.

Литература

- [1] *Дмитрий Левин*, *Hasher: технология безопасной сборки пакетов* // Первая международная конференция разработчиков свободных программ на Протве. Тезисы докладов. М., 2004. С.28–30.
- [2] *Дмитрий Левин*, *От SRPMS к GEAR* // Четвёртая международная конференция разработчиков свободных программ на Протве. Тезисы докладов. М., 2007. С. 14–18.
- [3] *Алексей Турбин*, *Комплементарное хеширование подмножеств* // Седьмая конференция разработчиков свободных программ. Тезисы докладов. М., 2010. С. 63–66.
- [4] *Михаил Шигорин*, *Макраме из дистрибутивов: mkimage-profiles* // Девятая конференция разработчиков свободных программ. Тезисы докладов. М., 2012. С. 48–49

Алексей Драгунов, Константин Драгунов, Леонид Гладченко
Псков, ГБУ Псковской области «Региональный центр информационных технологий», ФГБОУ ВО «МГТУ «СТАНКИН», Санкт-Петербургский политехнический университет Петра Великого

Создание открытой образовательной среды на базе операционных систем «Альт»

Аннотация

В статье рассматривается рамка Open Source-проекта, реализуемого Региональным центром информационных технологий Псковской области при поддержке «Базальт СПО» для обеспечения эффективного использования результатов национальных проектов в сфере образования и цифровизации.

Ключевые слова: *свободные цифровые инструменты, цифровая дидактика.*

Вступление

В Российской Федерации реализуются национальные проекты «Образование» и «Цифровая экономика», направленные на развитие, в том числе, ИТ — инфраструктуры и технологий в сфере образования. Результаты этих проектов создают возможности для развития обоснованного использования цифровых инструментов в работе образовательных организаций для расширения возможностей педагогов и учащихся. Мы не рассматриваем цифровые технологии как замену педагогу или «протез», позволяющий компенсировать недостатки, включая кадровые проблемы в школе. Скорее наоборот, использование «цифры» предъявляет новые требования к педагогу. Благодаря цифровым технологиям в сочетании с классической классно-урочной системой и сохраняя её преимущества, можно реализовать персонализированную работу с каждым учащимся, адаптировать темп обучения и уровень материала под особенности каждого, сократить время, затрачиваемое на рутину, связанную с проверкой работ и освободив его для основных задач школы — обучения и воспитания. «Цифра» позволяет своевременно выявлять проблемы и предвидеть их, помогать учителю — учить, а каждому ученику — учиться, вовлекать родителей в процесс обучения и воспитания.

Основная часть

Целью проекта является создание с использованием оборудования, поставленного в образовательные организации Псковской области открытой виртуальной образовательной среды, обеспечивающей следующий функционал:

1. Сервисы для организации работы школ на базе серверов ЦОС, объединённых в единую инфраструктуру в двух географических распределённых ЦОД в Псковской области (в настоящее время запущено 42 сервера):
 - a. Файловое хранилище, общий объём около петабайта. Каждой школе выделен диск размером не меньше 1 ТВ для совместной работы педагогов и учащихся¹.
 - b. Платформа для совместной работы с файлами и коллаборации пользователей — NextCloud².
 - c. Редакторы для совместной работы, интегрированные с NextCloud — OnlyOffice: текстовый редактор, табличный редактор, редактор презентаций, редактор pdf³; редактор диаграмм draw.io⁴;
 - d. Dolibarr — opensource CRM/ERP-платформа, которая может быть полезна образовательным организациям для обеспечения учёта, в том числе при организации внебюджетной деятельности. Рассматривается также несколько lowcode-платформ с точки зрения возможности их включения в число сервисов ЦОС для улучшения управленческих возможностей и расширения функционала РГИС «Цифровое образование Псковской области».
2. Среда для виртуализации учебных рабочих мест:
 - a. Управление виртуальными рабочими местами — OpenUDS⁵;
 - b. Обеспечение web-доступа к инфраструктуре рабочих столов — Apache Guacamole⁶.

¹<https://github.com/ceph/ceph>

²<https://github.com/nextcloud>, <https://www.altlinux.org/Nextcloud>

³<https://github.com/ONLYOFFICE>

⁴<https://github.com/jgraph/drawio>

⁵<https://github.com/VirtualCable/openuds>, <https://www.altlinux.org/VDI/OpenUDS>

⁶<https://github.com/apache/guacamole-server>, <https://www.altlinux.org/Guacamole>

3. Система управления цифровизацией образования региона

- а. Мониторинг и инвентаризация оборудования, программного обеспечения и функционирования ЕСПД — собственная разработка⁷

4. Система проведения занятий и виртуальная студия:

- а. Аппаратно-программный комплекс учебного класса, включающий в себя в части оборудования в максимальном варианте: микросервер на базе неттопа или ноутбука, качественные web-камеры и/или ip-камеры (две штуки), графический планшет и/или стеклянная сенсорная доска, интерактивные планшеты высокого разрешения, оборудование учащихся (ноутбуки или планшеты, либо смартфоны), петличный микрофон.
- б. Программное обеспечение для управления уроком: камерами и микросервером с **Open Broadcast Studio — OBS**, а также выводом виртуальных машин в **Guacamole**. Разрабатывается с использованием возможностей управления OBS с использованием WebSocet, а также управления IP-камерами по протоколу ONVIF⁸ и web-камерами с использованием Video4Linux⁹.
- в. Видеосервер с поддержкой низколатентного стриминга сцен из **OBS** в браузер по протоколу WHIP¹⁰.

5. Интеграционное программное обеспечение — **Keycloak**¹¹ — идентификация и аутентификация пользователей.

Рабочее название проекта «Открытая образовательная среда» (open educational environment). Планируется, что решение будет интегрировано с РГИС «Цифровое образование Псковской области» и ФГИС «Моя школа», в частности — за счёт использования единой аутентификации пользователей с использованием **keycloak**, связанной с модулем аутентификации РГИС, в свою очередь — использующей ЕСИА для доступа ко всем сервисам сферы образования, включая электронные журналы и дневники.

⁷<https://git.integricks.ru>

⁸<https://github.com/KuroNeko-san/ponvif>

⁹<https://github.com/mcqueeney/v4l2-logitech-camera-control-scripts>

¹⁰<https://github.com/glimesh/broadcast-box-tab-readme-ov-file-what-is-broadcast-box>, <https://github.com/AirenSoft/OvenMediaEngine>

¹¹<https://github.com/keycloak/keycloak>

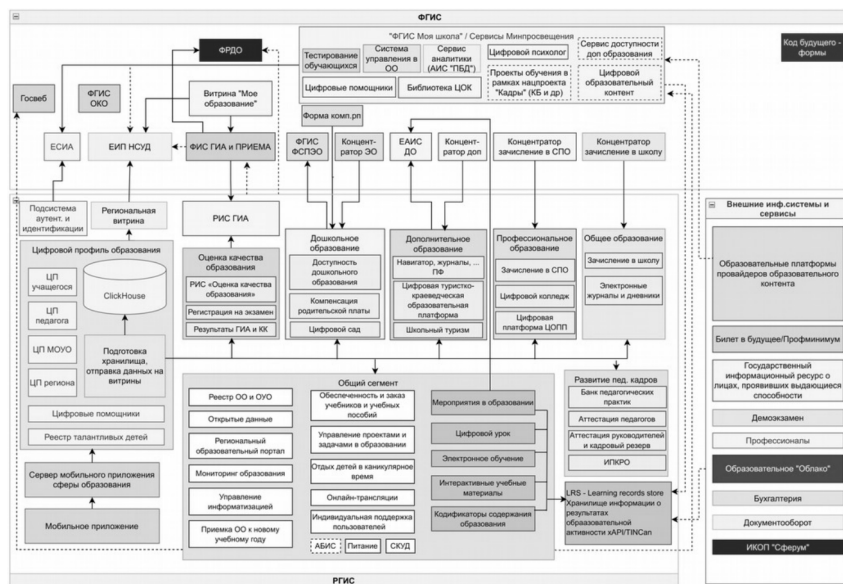


Рис. 1: РГИС «Цифровое образование Псковской области»

Рисунок 1 иллюстрирует структуру и основной функционал системы. Проведённые нами предварительные эксперименты показали возможность создания решения с использованием выбранных технологий и свободного программного обеспечения. Предполагается, что в качестве коммуникационного сервиса будет использован Сферум/ВКМ, входящий в состав ФГИС «Моя школа». При испытаниях прототипа мы использовали вывод сцены из OBS с использованием виртуальной web-камеры, однако стало понятно, что в этом случае качество видео, уходящего в Сферум достаточно низкое. В случае просмотра такой камеры локально, с трансляцией окна через webrtc в конференцию Сферум удалось добиться передачи изображения без потери качества.

Никита Шалаев

Санкт-Петербург, ФМОПИ СЗИУ РАНХиГС

Язык R и высшая математика для студентов-политологов: опыт построения практико-ориентированного курса

Аннотация

Студенты-политологи (как и в целом студенты социальных наук) обычно относятся к изучению высшей математики как к не нужной трате времени, поскольку перспектив применения этих навыков они не видят, а преподавание строится вокруг классического подхода «с бумагой и карандашом». Этот доклад будет посвящён попытке изменить ситуацию в более продуктивную сторону через совмещение преподавания математики и изучения языка R и среды разработки RStudio. В рамках доклада будет рассмотрен опыт организации курса высшей математики, построенного на базе свободного и бесплатного программного обеспечения и ориентированного на решение практических задач, и интеграции его в процесс обучения в целом.

Ключевые слова: *математика, язык R, социальные науки.*

Следует отметить, что вопрос «особого» подхода к обучению математике студентов по специальностям, относящимся к социальным наукам, возник не сегодня. Идея о том, что в данном случае нужен некоторым образом адаптированный курс, была реализована, к примеру, в учебнике Акимова (МГИМО) [1], О'Брайена и Гарсии (Саутгемптонский университет) [5], Гилла (Калифорнийский университет в Дейвисе) [3] и других. Однако обычно речь идёт об отборе конкретных областей математики, с которыми в социальных науках шанс встретиться выше. В остальном же преподавание построено вокруг классической модели обучения математике самой по себе, хотя по возможности и с приведением релевантных примеров из социальных наук (в основном, правда, экономики).

В то же время идея необходимости математики в социальных науках не вызывает сомнения, как бы ни казалось студентам, что выбранные ими специальности представляют собой тихие гавани, убежища, свободные от формул и вычислений. Здесь можно опереться на авторитетное мнение политолога Рейна Таагенперы [6], с точки зрения которого математика должна быть поставлена во главу угла при

проведении исследований, в то время как довольствоваться математикой, «законсервированной» внутри готовых статистических процедур разных прикладных программ, есть путь глубоко пагубный и скорее вредящий прогрессу в исследованиях, чем способствующий ему — и при этом совершенно типичный в наше время.

Существуют, однако, и попытки сделать математику ещё «ближе» для студентов. Здесь можно упомянуть учебник Ахтямова (БашГУ) [2], где параллельно с подачей «классического» материала рассматривалась работа и решение задач с системой компьютерной алгебры *Maple*. Этот подход можно покритиковать разве что за выбор проприетарного (и довольно дорогого) программного обеспечения для нужд образовательного процесса. Однако все вышеперечисленные идеи легли в основание курса, являющегося предметом доклада. С одной стороны, в рамках этого курса особое внимание было уделено областям математики, которые представляют наибольший практический интерес для политологов (например, теория множеств и булева алгебра, и, как следствие, качественный сравнительный анализ — QCA; разностные уравнения; экстраполяция и интерполяция данных). С другой стороны, для решения математических проблем было привлечено программное обеспечение — но уже свободное и бесплатное: язык R и среда разработки RStudio, а также несколько «опорных» пакетов из арсенала CRAN (прежде всего, *Ryacas* и *rgsma*).

Выбор R неслучаен. Во-первых, он широко используется в настоящих, полноценных исследованиях в социальных науках по своему прямому назначению — как язык статистического программирования. Поэтому изучая его, студенты изучают нечто, заведомо им полезное и в других областях профессиональной деятельности. Это выгодно отличает R от более пригодных именно к математическим задачам систем компьютерной алгебры (например, *Maxima*) или численного анализа (скажем, *GNU Octave*). Кроме того, экосистема R (прежде всего, репозиторий CRAN) постоянно прирастают новыми пакетами, которые реализуют новейшие идеи и методы, позволяя пользователю оставаться на передовом крае науки. Во-вторых, R и RStudio являются свободным и бесплатным программным обеспечением, что выгодно отличает их от проприетарных продуктов (*Maple*, *Matlab*, *SPSS*, *STATA* и т. д.), доступ к которым может оказаться потерян в любой момент по множеству причин, а в нормальных условиях потребует от студентов дополнительных трат после окончания обучения. R же останется со студентами и после выпуска, безо всяких условий и ограничений,

что в целом отвечает критерию фундаментальности и постоянства получаемых в стенах университета знаний. Наконец, в-третьих, R и RStudio позволяют реализовывать концепции «грамотного программирования» и воспроизводимых исследований, в том числе в коллективе.

Конечно, в рамках курса пришлось пойти и на некоторые компромиссы. Так, например, был сделан выбор в пользу системы компьютерной алгебры *yacas*, так как она легко доступна в виде одного пакета (*Ryacas*), в то время как гораздо более мощная *Maxima* требует отдельной установки. Также большое внимание уделяется не только «первосортным» аналитическим методам и компьютерной алгебре, но и численным методам решения задач, для которых R приспособлен много лучше [4].

С другой стороны, выявились и неожиданные преимущества от реализуемой схемы. Например, списывание оказалось легко выявляемым как при копировании кода между студентами, так и при выдаче им за свой труд кода, написанного кем-то ещё (возможно, как раз знакомым студентом какого-то технического направления). И в целом по результатам тестирования ФЭПО отмечается последовательное улучшение результатов. Наконец, благодаря широким возможностям R, эту же программную основу получилось повторно использовать и в других курсах: от теории игр до анализа государственной политики.

Литература

- [1] *Акимов В. П.* Математика для политологов. Учебное пособие. 2 изд. М.: МГИМО-Университет, 2011.
- [2] *Ахтямов А. М.* Математика для социологов и экономистов. Учебное пособие. 2 изд. М.: Физматлит, 2008.
- [3] *Gill J.* Essential Mathematics for Political and Social Research, Cambridge University Press, 2006.
- [4] *Howard James P.* Computational Methods for Numerical Analysis with R. CRC Press (Taylor & Francis group), 2017.
- [5] *O'Brien R. J., Garcia G. G.* Mathematics for Economicst and Social Scientists. Palgrave Macmillan, 1971.
- [6] *Taagepera R.* Making Social Sciences More Scientific: The Need for Predictive Models. Oxford University Press, 2008.

Георгий Курячий

Москва, ВМК МГУ

Проект: UNIX <http://uneex.org>

Курс «Сети в Linux»: опыт пяти лет преподавания

Аннотация

Рассказ о том, как преподаётся на факультете курс «Сетевые протоколы в Linux», как устроен образ виртуальной машины для выполнения домашних заданий и какие дополнительные инструменты используются при их решении и на лекции.

Ключевые слова: *virtualbox, преподавание, сети, ALT.*

У человека, даже получившего технологическое образование, тема «компьютерные сети» зачастую разваливается прямо в голове на три слабо пересекающиеся области: теория компьютерных сетей, которую ему рассказывали на лекциях, практика настройки отдельных параметров (как минимум персонального компьютера — или, если образование было достаточно продвинутым, какого-нибудь сетевого устройства на нескольких типичных задачах), и, наконец, махровая мифология, заполняющая довольно обширные промежутки между теорией и практикой.

Работая системным администратором на факультете ВМК, где мне приходилось интегрировать и аппаратные устройства, и компьютеры под управлением FreeBSD, Solaris и Linux, я убедился, что ни первая, ни вторая, ни тем более — третья области мне особо не помогают. Тогда-то и зародилась идея об учебном курсе, ведущем от теории к практике, но реализована она была несколько позже.

Попытка №0 была предпринята почти 20 лет назад — это был довольно «классический» курс по сетям, в котором я делился (бегая у меловой доски) своими тогдашними админскими знаниями[1]. За основу был взят стек протоколов TCP/IP, изложение шло по нему снизу вверх, от аппаратного уровня до прикладного, на котором мы задерживались, изучая отдельно DNS, электронную почту и борьбу со спамом в ней. Отчётом был устный экзамен, который принимал не только я, но и мои коллеги-сетевики. Думаю, что для того времени это был хороший, годный спецкурс.

Следующая попытка — 2013 года[2] — уже включала в себя решение домашних заданий для двух виртуальных машин под VirtualBox

и полуавтоматическую их проверку, и вместо почты мы занимались межсетевыми экранами и `secure shell`. Ресурсов на этот курс было потрачено много, перезапустить его я не решился.

В 2020 году произошёл курьёз. Мой коллега жалуется мне: в его спецкурсе «Архитектура роутеров» (или как-то так) слишком много обзорного и нет практики. Не соглашусь ли я прочесть пару лекций про маршрутизацию в Linux? Сговорились на 4 часа, два по L2, два — по L3. Под конец семестра выяснилось, что только по этим темам возможны практические задания, так что пришлось их тоже подготовить [3] — на этот раз с запуском пяти виртуальных машин с вариацией их функций и топологии сети.

В 2022 году я решил «пришить к этой пуговице пару брюк». Получившийся курс воспроизводит структуру, заданную в 2013 году — вводная лекция, по две лекции на каждый уровень TCP/IP, разговор о средствах перманентной настройки, туннелирования и о межсетевых экранах. Обязательные домашние задания включают в себя развёртывание некоторой сетевой топологии из клонов эталонного образа виртуальной машины для VirtualBox, настройку заданных параметров и пересылку т. н. «отчётов» — записи диалога командной строки в консоли каждой из виртуалок. Отчёты следует посылать по электронной почте на выделенный адрес. В 2024 году проверка этих отчётов стала автоматической [4].

Лекции читаются аудиторно и сопровождаются стримингом в YouTube. Проверкой домашних заданий занимается специальное приложение HWorker [5], написанное студентами ВМК в рамках летней стажировки в «Базальт СПО». Образ виртуальной машины сделан на базе минимальной сборки ALT Starterkit JEOS [6], в который вручную установлены требуемые пакеты. Изменений в настройках ОС практически нет, за исключением явного использования загрузчика и консоли на последовательном порту, подключение к которой облегчает копирование текстовых данных. Дополнительно написано несколько `shell-утилит` [7], позволяющих со стороны хост-системы унифицированно управлять клонированием виртуалок и привязкой их к виртуальным сетям, и `shell-утилит`, позволяющих в гостевой системе «на скорую руку» настраивать сеть, имя компьютера, тип терминала и т. п.

Из ближайших планов — перевод образа на автоматизированную сборку (по аналогии с ALT Starterkit), из среднесрочных — адаптация образа для использования в студенческих работах (развёртыва-

ние площадки с заданной топологией и запуск там чего-нибудь дипломного) и в связи с этим доработка системы передачи настроек сети.

Литература

- [1] <https://uneex.org/LecturesCMC/TCPIP2004>
- [2] <https://uneex.org/LecturesCMC/LinuxNetwork2013>
- [3] <https://uneex.org/LecturesCMC/LinuxNetwork2020>
- [4] <https://uneex.org/LecturesCMC/LinuxNetwork2024>
- [5] <https://github.com/FrBrGeorge/HWorker>
- [6] <https://www.altlinux.org/Starterkits/JeOS>
- [7] <https://github.com/FrBrGeorge/vbsnap>

Александр Смирновский

Санкт-Петербург, Санкт-Петербургский политехнический университет Петра Великого

Программирование для непрограммистов: какой язык программирования учить специалистам по численному моделированию?

Аннотация

В докладе представляется опыт преподавания программирования на ЯП Python студентам первого курса бакалавриата Физико-механического института политехнического университета по направлению «Прикладные математика и физика». Основной вопрос, на который до сих пор нет окончательного ответа, — какой язык программирования следует использовать в обучении студентов по данной специальности на протяжении всего периода обучения? Размышления на эту тему и посвящён данный доклад.

Ключевые слова: *программирование, численное моделирование, Python*

На протяжении уже трёх лет студентам первого курса бакалавриата Физико-механического института Политехнического университета по направлению «Прикладные математика и физика» читается

курс, который в рамках учебной программы гордо именуется «Цифровой практикum» (первый семестр), а во втором семестре имеет более осмысленно название «Практикum по языкам программирования». Данные курсы возникли в результате «трансформации» учебного плана всего университета с целью увеличения числа и некоторой унификации «цифровых» курсов, под которыми понимается не только (и не столько) программирование, но и такие малопонятные абитуриенту (да и преподавателям, впрочем, тоже) словосочетания, как: «цифровая культура», «цифровая грамотность», «системы искусственного интеллекта» и пр. Учебные планы некоторых курсов унифицированы и преподаются студентам в большей или меньшей степени централизованно, тогда как наполнение других курсов полностью определяет то подразделение Политехнического университета, преподаватели которого и ведут занятия.

Что касается упомянутых курсов «Практикum по языкам программирования» и «Цифровой практикum», то его наполнением и преподаванием занимаются сотрудники Высшей школы прикладной математики и вычислительной физики Физико-механического института. Для дальнейшего обсуждения вынесенной в заголовок темы доклада необходимо сделать небольшое отступление и некоторый исторический экскурс.

Как указано на сайте Физико-механического института СПбПУ [1], образовательная программа по направлению «Прикладные математика и физика» (направленность «Математические модели и вычислительные технологии в гидроаэродинамике и теплофизике») нацелена на подготовку бакалавров, обладающих обширными знаниями в области физико-математических наук, преимущественно ориентированных на дальнейшее обучение в магистратуре и обладающих практическими навыками математического и экспериментального моделирования при решении задач гидроаэродинамики, теплофизики, механики деформируемого твёрдого тела и смежных научно-технических областей. Общая концепция подготовки основана на сочетании фундаментальных дисциплин математической и физико-механической направленности с курсами по современным вычислительным и информационным технологиям, применяемым для решения общенаучных и прикладных задач физики и механики сплошных сред. Таким образом, «цифровой» аспект является хоть и не основной целью обучения по указанной программе, но очень важной составляющей: на протяжении всего периода обучения студентам необходимо выполнять

различные задания, связанными с программированием, в том числе самостоятельно разрабатывать программы, реализующих численные алгоритмы для решения задач механики сплошных сред (гидроаэродинамики).

На протяжении многих лет основным инструментом для написания программ в процессе обучения в Политехническом университете был и остаётся язык **Fortran**, позволяющий с относительно небольшими усилиями реализовывать различные алгоритмы вычислительной физики, включая: методы Рунге-Кутты для решения ОДУ, численные методы для решения уравнения конвективного переноса скалярной величины, конечно-разностные методы аппроксимации уравнения теплопроводности и др. Более сорока лет назад преподавателями тогда ещё физико-механического факультета Ленинградского политехнического института балы разработана учебная программа, включающая как изучение ЯП **Fortran**, так и использование его для программирования в рамках различных курсов по численным методам. В 2002 году было издано учебное пособие [2], содержащее в том числе материалы по ЯП **Fortran**, которое используется студентами и преподавателями Политехнического университета и по настоящее время.

Таким образом, с конца 1980-х годов сформировался единый подход к преподаванию курсов для студентов специальности «Прикладные математика и физика», в котором на первое место ставится изучение физики, а также численных методов и подходов для решения физических задач, тогда как программирование является только вспомогательным инструментом. В этой связи, учебные курсы непосредственно по программированию предусмотрены по сути только на первом курсе бакалавриата.

С другой стороны, уже на протяжении многих лет внутри преподавательского коллектива продолжается дискуссия о том, что и как именно необходимо преподавать в рамках курсов по программированию. Эти дискуссии обусловлены следующими обстоятельствами.

Со стороны студентов есть запрос на обучение программированию, которое было бы не только современно, но и могло бы пригодиться в различных областях профессиональной деятельности. Не раз со стороны некоторых студентов возникал вопрос, почему им до сих пор приходится изучать **Fortran**, а не более современные и распространённые ЯП? Конечно, с учётом современных тенденций (в мае 2024 году **Fortran** вошёл в ТОП-10 популярных ЯП в рейтинге TIOBE [3]),

совсем отказываться от **Fortran** было бы нецелесообразно, особенно если учесть, что в области НРС (High Performance Computing) и вычислительных методов он является довольно удобным и относительно простым, но в тоже время мощным инструментом (особенно по сравнению с **C** и даже **C++**), поддерживающим «из коробки» параллелизацию OpenMP, для него доступны библиотеки MPI и многое другое из области НРС. Тем не менее, недовольство студентов необходимостью изучения **Fortran** по-прежнему остаётся.

С другой стороны, как показывает ежегодный опрос студентов первого курса бакалавриата, который проводится на протяжении нескольких лет в начале сентября, примерно от трети до половины студентов оценивают свой навык программирования на низком уровне, а часть из них в целом относятся к программированию скорее негативно. То есть складывается ситуация, когда часть студентов не готова и даже в принципе не хочет заниматься программированием. Таким образом, задача преподавателей, преподающих программирование, в том числе заключается в том, чтобы заинтересовать и вовлечь студентов в область, связанную с программированием, а также «поднять» уровень тех из них, кто приходит из школы с недостаточными знаниями программирования, чтобы на старших курсах студенты могли без проблем писать программы, необходимые для выполнения заданий.

И здесь возникает ещё одно важное обстоятельство: многие преподаватели и сами не видят интереса и целесообразности преподавать ЯП **Fortran**, а мотивация преподавателей, как известно, в настоящее время практически единственный способ хоть какого-то развития высшей школы в РФ...

И последнее обстоятельство (last but not least) — поскольку специальность «Прикладные математика и физика» включает не только вычислительную составляющую, но и такие важные аспекты, как экспериментальные исследования и обработка данных (лабораторные работы), то студентам необходимы соответствующий программный инструментарий. В рамках «цифровых» курсов студенты изучают различное ПО для подготовки графиков, написания текстов и пр. (в основном свободное программное обеспечение, включая L^AT_EX, SciDaVis и др.). Хорошим подспорьем для них является также и знание языков программирования, позволяющих автоматизировать процессы обработки данных и их анализ.

Таким образом, складывается ситуация, когда, с одной стороны, учебная специальность «далека» от канонического ИТ, а с другой — программирование является важной составляющей как в обучении, так и для дальнейшей профессиональной деятельности. И во весь рост встаёт вопрос, вынесенный в заголовок доклада: так какой же язык программирования следует учить специалистам, чья основная область деятельности — вычислительная прикладная физика? Как обычно бывает в подобных ситуациях, не существует «серебряной пули» — одного «универсального» ЯП, позволяющего и обрабатывать данные, и создавать вспомогательные инструменты для их анализа, и писать высокопроизводительные программы (в том числе для НРС). В настоящее время такую нишу пытается занять ЯП *Julia* [4], который совмещает одновременно и возможность написания высокопроизводительного кода, и простоту и наглядность для написания простых программ для обработки и анализа данных. К сожалению, несмотря на весьма быстрое развитие языка и его инфраструктуры, в настоящий момент имеется очень мало как различных учебных материалов на русском языке, так и опыта его использования в преподавательской среде. Весьма вероятно, что этот язык прочно займёт свою нишу в области научных вычислений и станет там весьма распространён, но скорее всего не в ближайшем будущем.

Таким образом, текущие реалии таковы, что наиболее «подходящим» ЯП для обучения, а также для обработки и анализа и данных, был и остаётся *Python*. Именно по этой причине этот язык был выбран в качестве ЯП для обучения студентов, поступающих на направление «Прикладные математика и физика». ЯП *Python* преподаётся в течение двух первых семестров первого курса бакалавриата. Для обучения созданию высокопроизводительных программ с возможностью параллелизации (*OpenMP* и *MPI*) по-прежнему используется *Fortran*, который в настоящее время преподаётся в рамках второго семестра первого курса.

Пока ещё рано говорить, насколько данная концепция преподавания программирования является удачной в контексте обучения по специальности, связанной с численным моделированием физических процессов, но по крайней мере большинство студентов весьма положительно оценивают приобретённый опыт программирования на *Python*.

Литература

- [1] Официальный сайт Физико-механического института с описанием направления подготовки «Прикладная математика и физика», https://physmech.spbstu.ru/edu/03.03.01/03.03.01_01/
- [2] Шелест В. Д., Программирование: учебное пособие, 2022
- [3] Индекс ТЮВЕ популярности языков программирования (май 2024), <https://www.tiobe.com/tiobe-index/>
- [4] Язык программирования Julia, <https://julialang.org/>

Сергей Быков, Алина Малетина, Лидия Корунова
Москва, МГТУ им. Н. Э. Баумана

Использование свободного ПО в обучении по направлению «Бизнес-информатика»

Аннотация

Рассматривается текущее положение свободного ПО в программах обучения по направлению «Бизнес-информатика» ведущих вузов России.

Ключевые слова: *бизнес-информатика, диаграммы, business-informatics, diagrams.*

Эксперты отмечают высокую востребованность ИТ-специальностей в компаниях, многие из которых находятся на стыке информационных технологий и бизнеса [1]. Поэтому направление «Бизнес-информатика» является популярным для предприятий. Это направление привлекает внимание не только бизнеса, но и государства. Например, новая программа бакалавриата «РЕСУРС России» в РАН-ХиГС включает два направления: «Бизнес-информатика» и «Государственное и муниципальное управление». Также множество национальных проектов связаны с цифровой трансформацией и цифровой экономикой, требуя понимания как технической, так и бизнес-составляющей.

Студенты проявляют интерес к «Бизнес-информатике», как видно по динамике проходных баллов в ведущих российских университетах. Знания, получаемые по этому направлению, должны быть актуальными и применимыми в реальной жизни.

Для анализа программ обучения был взят «Топ 10 вузов цифровой экономики», составленный АНО «Цифровая экономика» при поддержке Минцифры и Ассоциации предприятий компьютерных и информационных технологий [2]. Были рассмотрены программы бакалавриата, поэтому в итоговый список вузов вошли: МГТУ им. Н.Э.Баумана, НИУ ВШЭ, ИТМО, МИФИ, СПбГУ, МИРЭА, СПбПУ Петра Великого, СПбГУТ имени профессора М.А.Бонч-Буревича и МЭИ.

Для оценки опыта студентов выбранных вузов провели опрос у будущих специалистов по направлению подготовки 38.03.05 «Бизнес-информатика». Результаты опроса показывают, что культура применения СПО слабо развита, а студенты часто не осведомлены о свободных альтернативах программам. В некоторых вузах есть конкретные требования к использованию определённого ПО, но в большинстве случаев они отсутствуют. При этом половина опрошенных за время учёбы сталкивается с проблемой неудобного и нелогичного ПО, которое приходится использовать. Проблему можно решить за счёт использования СПО.

Для её решения были проанализированы учебные планы и выделены группы предметов, схожие для всех направлений обучения:

- Программирование и разработка.
- Экономика и финансы.
- Управление и бизнес-анализ.
- Информационные системы и технологии.

В табл. 1 рассмотрено количество дисциплин, входящих в каждую из групп. В случаях, когда в университете было несколько программ по направлению «Бизнес-информатика» (например, СПбГУТ им. М.А.Бонч-Буревича), количество дисциплин суммировалось по группам дисциплин.

Сложив количество дисциплин по группам, увидим, что больше всего учебных предметов в группе «Управление и бизнес-анализ», а меньше всего в группе «Программирование и разработка». Исходя из этого можем говорить о том, в каких группах дисциплин стоит развивать использование СПО.

Группа дисциплин «Программирование и разработка» нацелена на развитие практических навыков создания программного обеспечения и веб-приложений. В этой группе уже используются языки программирования, для которых существуют свободные компиляторы,

ВУЗ	Программирование	Экономика	Управление	ИТ
МГТУ им. Баумана	12	6	6	7
НИУ ВШЭ	5	8	17	4
ИТМО	11	5	5	9
МИФИ	9	6	10	7
СПбГУ	5	6	6	6
МИРЭА	4	4	7	7
СПбПУ Петра Великого	5	34	47	32
СПбГУТ им. М.А. Бонч-Буревича	5	21	34	24
МЭИ	6	11	4	4
Σ	62	101	136	100

Таблица 1: Количество учебных дисциплин по группам.

поэтому стоит предлагать студентам среды разработки с открытым исходным кодом, например, PyCharm, VS Code или Vim и Emacs для продвинутых студентов.

Группа дисциплин «Экономика и финансы» направлена на обучение навыкам, связанным с экономическими дисциплинами. Несмотря на то, что многие теоретические дисциплины не требуют специфического ПО, для практического применения экономических моделей и анализа данных студенты могут использовать программы с открытым исходным кодом, такие как R и Python Pandas, matplotlib для анализа данных; Tableau, Power BI для визуализации данных и LibreOffice для создания электронных таблиц. SPSS, SAS могут использоваться для анализа и обработки статистических данных, построения прогнозов и моделей исследований.

В группе дисциплин «Управление и бизнес-анализ» студенты изучают принципы эффективного управления бизнесом и разрабатывают стратегии развития компаний. С помощью СПО Camunda можно анализировать, оптимизировать и автоматизировать бизнес-процессы в организации. Draw.io, Dia, Miro, PlantUML позволяют моделировать бизнес-процессы. DecisionTools Suite, LINDO, SolverStudio можно использовать для решения задач оптимизации, принятия решений и разработки стратегий, OpenProject и GanttProject — для управления проектами.

При изучении группы дисциплин «Информационные системы и технологии» студенты осваивают принципы проектирования и разработки информационных систем, управляют сетевыми ресурсами и занимаются разработкой и внедрением новых технологий. Решениями могут стать такие СПО, как PostgreSQL и MySQL для создания, управления, анализа и оптимизации реляционных баз данных, написания

запросов, моделирования структур данных, Bizagi для моделирования бизнес-процессов, анализа данных, оптимизации операций компании, Kali Linux, Metasploit, Wireshark — для анализа безопасности сети, сканирования уязвимостей и тестирования на проникновение. Кроме того, для изучения компьютерных сетей можно использовать GNS3.

Таким образом, была доказана актуальность использования СПО в обучении по направлению «Бизнес-информатика», показана необходимость развития культуры FOSS в этом направлении, предложены примеры свободного ПО для учебных дисциплин с опорой на анализ планов университетов.

Литература

- [1] А. Узбекова Конкуренция за ИТ-специалистов высокого класса обостряется, <https://rg.ru/2024/05/21/programmist-rastet-v-cene.html> Дата посещения 06.06.2024.
- [2] Рейтинг вузов цифровой экономики, <https://d-economy.ru/analitic/rejting-vuzov-cifrovoj-jekonomiki-2023/>, Дата посещения 06.06.2024.

Андрей Михеев

Москва

Проект: RunaWFE Free <http://runawfe.org>, <http://runawfe.ru>

Курсовые работы и производственная практика, связанные с проектами СПО, в Финансовом университете и НИУ ВШЭ

Аннотация

В докладе рассматриваются как традиционный, так и новый подходы к привлечению студентов в СПО-проекты. При традиционном подходе студенты разрабатывают небольшую часть функциональности для давно существующего СПО-проекта. В этом случае они могут увидеть результат своей деятельности в доступном всем желающим работающем программном продукте, показать другим разработчикам (а также возможному работодателю) хорошее качество своего кода и интересные технические решения.

Новый подход связан с подготовкой студентов к самостоятельной деятельности не только в области программирования, но и в бизнесе. В этом случае ищутся ниши в бизнесе, в которых за счёт не существовавших ранее условий или технологий студенты реализуют быстрые, технически не очень сложные программные решения, которые могут быть в том или ином виде востребованы на рынке. В данной работе студентам было предложено три направления и сделана попытка организации преемственности этих работ через СПО-проекты, в которых студенты следующих курсов в своих курсовых работах продолжают работы студентов предыдущих курсов, а возможные бизнес-стратегии предполагают косвенное использование СПО-проектов, связанное с востребованностью разрабатываемых решений для потенциальных пользователей.

Ключевые слова: *СПО, курсовые работы, производственная практика.*

Участие студентов в существующем проекте

В этом случае проекты производственной практики и курсовые работы, руководителем которых является автор настоящего доклада, соответствуют разработке небольшой законченной функциональности для свободной BRM-платформы с открытым исходным кодом [1]. Часто такие проекты связаны с реализацией какого-то элемента спецификации BRMN (например, «ветвление по данным», «бизнес-правило», «отложенный выбор», «подпроцесс-цикл» и т. п.) или с разработкой алгоритмов анализа графа бизнес-процесса (таких, как «определение недостижимых элементов бизнес-процесса», «определение возможных мгновенных циклов», «определение возможных ситуаций, в которых бизнес-процесс не сможет завершиться» и т. п.). В первом случае шире набор используемых студентами технологий, так как реализовать надо как клиентскую, так и серверную часть решения. Во втором случае — технологий надо изучать меньше, но сложнее алгоритмическая часть: требуется придумывать не использовавшиеся ранее алгоритмы. Использование СПО заметно облегчает как выполнение, так и оформление работ: не требуется заключать NDA, документация разработчика является полностью доступной, студентам не требуется давать права доступа к закрытым репозиториям, студенты могут передать разработанный код комиссии по защите курсовых, а также включить в пояснительную записку. Кроме того, в этом случае у студентов есть возможность показать разработанный код будущему работодателю.

Работы студентов, связанные с существующим СПО-проектом, ведутся в рамках учебного процесса НИУ ВШЭ.

Новые студенческие СПО-проекты

Новые студенческие СПО-проекты, организуемые автором настоящего доклада, связаны с движением «Стартап как диплом» и организацией групп ВТСК (временных творческих студенческих коллективов) в Финансовом университете.

В современных вузах кроме традиционного обучения постепенно появляются формы подготовки студентов, связанные с предпринимательской деятельностью. В случае студенческих стартапов учащиеся ищут ниши в бизнесе, в которых они могут предложить востребованные решения. Часто это происходит в областях, в которых недавно появились инновационные технологические решения или новые формы социальной организации (новые бытовые устройства, новые типы социальных сетей или средства коммуникации и т.п.). Студенты быстрее и легче адаптируются к изменившимся условиям, чем более старшие возрастные категории. Направление «Стартап как диплом» предполагает, что выпускник защищает реальный технологический проект, имеющий бизнес-перспективы. В рамках данного подхода была сделана попытка найти связанную с ИТ-разработкой предметную область, в которой за счёт достижений последнего времени для студентов появились возможности создавать востребованные бизнесом решения.

Было предложено три направления для курсовых работ, инициатива которых пришла из Московского государственного университета геодезии и картографии. Также была произведена попытка организовать преемственность проектов, при которой студенты старшего курса после завершения курсовой работы передают разработанный код студентам младшего курса. Подход «Стартап как диплом» предполагает, что студенты, организовавшие стартап, являются собственниками как компании, так и разрабатываемого программного продукта. Поэтому было предложено разрабатывать проекты под свободной лицензией, постепенно добавляя в участники проекта новых студентов. При этом любой студент может сделать на основе проекта MVP (минимально жизнеспособный продукт) собственного стартапа и иметь на него права собственности, как на составное произведение.

Предложенные направления для курсовых работ:

- Система определения местоположений объектов по прямым и обратным угловым засечкам, связанным с электронной картой.
- Система для работы с данными сервиса FIRMS, осуществляющая подписку на оповещения о пожарах, происходящих в указанном районе.
- Система, осуществляющая мониторинг объектов культурного наследия при помощи parsing-метода и сообщающая о состояниях реставрационных работ.

Приведём более подробное описание одного из проектов.

Система определения местоположений объектов по прямым и обратным угловым засечкам

Проект посвящён решению задачи определения координат наблюдаемых явлений на основе данных, которые одновременно вводят несколько пользователей, находящихся в разных местах, при помощи бытовых мобильных устройств без использования специального технологического оборудования. Такая система позволит использовать инициативу населения в упреждении нежелательных событий и предотвращении чрезвычайных ситуаций. Пользователями системы могут быть свидетели дистанционно наблюдаемого пожара, крушения, иных следов бедствия или опасного природного явления.

Способ определения местоположения наблюдаемого явления основан на его наблюдении из разных точек. Если объект внимания одновременно наблюдается с нескольких ракурсов, для определения его координат нужно засечь направления на объект из точек с известными координатами. Такой метод определения координат называется прямой угловой засечкой. Обратная угловая засечка применяется для определения координат наблюдателя, не знающего своего положения. В этом случае для получения координат засекаются направления из точки нахождения наблюдателя на объекты с известными координатами.

Система должна давать возможность одновременно наблюдающим явление пользователям, часть из которых знает свои координаты, а часть может не знать, ввести направления на явление и известные объекты местности, после чего система определит координаты явления, а также пользователей, не знающих свои координаты и отобразит их на карте. Ввод направлений может производиться явно

через построение линий на экране мобильного устройства, направленных на соответствующие объекты или путём фотографирования окружающей местности и указания на фотографии известных и неизвестных объектов (в этом случае система самостоятельно рассчитывает соответствующие углы).

По этой теме за два года написали курсовые работы несколько студентов. Пока нельзя сказать, что предполагающийся практический результат этой деятельности достигнут. Использовать разработанный код для реальной работы на местности ещё нельзя, т. к. при выполнении этих курсовых большинство студентов ориентируется скорее на получение оценки, чем на доведение приложения до практического использования. С преемственностью на практике тоже оказалось всё достаточно сложно. Однако, соответствующие проекты с открытым кодом есть. По результатам работы над одним из проектов была опубликована статья [2].

Выводы

В докладе рассмотрено привлечение студентов к разработке существующего СПО-проекта и попытка организации студенческих СПО-проектов в рамках новых форм работы со студентами, таких как направление «Стартап как диплом» и организация групп ВТСК. Если привлечение студентов к работе в существующем СПО-проекте хорошо отработано и имеет многолетнюю практику, то новые формы работы со студентами приживаются непросто. Однако, вузам эти направления интересны и можно ожидать, что практические результаты нового подхода через некоторое время тоже появятся.

Литература

- [1] Ссылки на сайты проекта RunaWFE Free: <http://runawfe.org>, <http://runawfe.ru>
- [2] *Михеев А. Г., Брылёв В. Д., Грузинов В. С., Савельев М. В.* Методика коллективной фиксации местоположения наблюдаемых чрезвычайных ситуаций методом прямой засечки на основе мобильного телекоммуникационного приложения с картографическим интерфейсом. «Системный анализ и прикладная информатика». 2023; (3): с. 40–46

Мария Полякова, Кирилл Чувилін
Москва, ООО «Открытая мобильная платформа»
<https://gitlab.com/omprussia/edu>

Образовательная среда «Аврора»

Аннотация

Доклад знакомит с опытом формирования академического сообщества вокруг технологий, используемых и создаваемых ООО «Открытая мобильная платформа» (ОМП), компанией-разработчиком мобильной ОС «Аврора». Приведён пример применения опенсорс-подхода к публикации учебно-методических материалов, разработанных ОМП и направленных на профильное обучение. Предложены варианты развития экосистемы академических и промышленных партнёров.

Ключевые слова: *ОС «Аврора», системная разработка, мобильная разработка, высшее образование, дополнительное образование.*

ООО «Открытая мобильная платформа» (ОМП) [1] является разработчиком средств, позволяющих выстроить доверенную инфраструктуру для работы с мобильными устройствами. Функционирование такой инфраструктуры обеспечивается, в том числе, операционной системой «Аврора» и платформой управления устройствами «Аврора Центр». ОС «Аврора» [2] предназначена для использования на смартфонах и планшетах корпоративными заказчиками и государственными компаниями.

Разработка операционной системы и прикладного ПО — это набор сложных технических задач, которые требуют высокой квалификации от сотрудников. При этом наш опыт и опыт наших партнёров показывает, что у многих кандидатов на открытые вакансии не достаёт фундаментальных знаний по ряду технологий, важных как для нас, так и для российского ИТ-сообщества в целом. Подготовка квалифицированных кадров вузами при поддержке ИТ-компаний сейчас позитивно скажется на укреплении технологического суверенитета уже в краткосрочной перспективе.

Мы запустили активности, направленные на создание академического сообщества вокруг технологий, с которыми мы работаем, и учебных курсов. Целевое видение результата — это экосистема, созданная партнёрством образовательных и промышленных организаций, принимающих непосредственное участие в развитии материалов, доступных под открытыми лицензиями.

Основные направления следующие:

- Сайт: информация о наших активностях, в т.ч. стажировки, сотрудничество по проведению исследованию, полезные ссылки [3].
- Открытые каналы и группы в Telegram [4].
- Портал для разработчиков [5], предоставляющий необходимые для самостоятельной разработки приложений информацию и ресурсы.
- Учебно-методические материалы на git-репозитории [6].

В нашей работе оказалось важным, что публикация материала учебного курса — это размещение РИД коммерческой компании в публичном доступе под пермиссивной лицензий. В этом есть много нюансов, начиная с организационных вопросов и учёта на балансе, заканчивая техническими аспектами. Для этого мы проработали следующие вопросы:

- Что публикуем — учебно-методические материалы по технологиям, используемым в ОМП, которые могут быть применены образовательными организациями для создания или расширения учебных курсов.
- На каких условиях предоставляем доступ — доступ предоставляется по условиям пермиссивной лицензии, допускающей, в том числе, коммерческое использование.
- На каких условиях принимаем правки — регламент правок определяется отдельным соглашением, которое, в том числе, управляет сохранностью лицензии.
- От кого ждём участие — преподаватели образовательных организаций, внедряющих курсы на основе наших учебно-методических материалов, разработчики открытого сообщества «Аврора».
- Где всё это делаем — в git-репозитории. На момент написания тезисов используется публичный GitLab.
- Какие инструменты, форматы, подходы используем — все используемые для подготовки и использования контента инструменты являются программным обеспечением с открытым исходным кодом и возможностью свободного их использования.

Литература

- [1] Сайт ООО «Открытая мобильная платформа»: официальный сайт — 2024г. — URL: <https://www.omp.ru> (дата обращения 06.06.2024)
- [2] Сайт «Аврора своя система»: официальный сайт — 2024г. — URL: <https://auroraos.ru> (дата обращения 06.06.2024)
- [3] Страница «Аврора образовательная среда»: официальный сайт — 2024г. — URL: <https://auroraos.ru/education> (дата обращения 06.06.2024)
- [4] Телеграмм-канал «Aurora Education» — URL: https://t.me/omp_edu (дата обращения 06.06.2024)
- [5] Сайт «Портал разработчика»: официальный сайт — 2024г. — URL: <https://developer.auroraos.ru> (дата обращения 06.06.2024)
- [6] Репозиторий Open Mobile Platform Ltd. — URL: <https://gitlab.com/omprussia/edu> (дата обращения 06.06.2024)

Иван Хахаев

Санкт-Петербург, СПбГЭТУ «ЛЭТИ»

Позиционирование свободных программ для моделирования в электронике

Аннотация

Рассматриваются программные средства со свободными лицензиями для моделирования в аналоговой и цифровой электронике в разрезе применимости в учебном процессе. Проведён анализ актуальности и функциональных возможностей 15 пакетов для аналоговой и цифровой электроники, предложены варианты замен распространённого иностранного программного обеспечения.

Ключевые слова: *аналоговая электроника, цифровая электроника, моделирование, SPICE, свободные программы.*

Перевод учебного процесса на отечественные дистрибутивы Linux даёт возможность использовать пакеты со свободными лицензиями для поддержки ключевых для СПбГЭТУ «ЛЭТИ» дисциплин в области электроники, радио- и электротехники, при этом нужно понять области применения и пригодность для решения задач разного уровня

(учебных или профессиональных). В докладе рассматриваются средства моделирования в аналоговой и цифровой электронике в разрезе применимости в учебном процессе.

Анализ пункта «Требования к программному обеспечению» раздела «Описание информационных технологий и материально-технической базы» рабочих программ по дисциплинам (РПД), содержащим в названиях ключевые слова «электроника», «электротехника», «радиотехника» и «схемотехника» по состоянию на декабрь 2023 г. показал, что только в 8 РПД из общего количества 128 в данном пункте указано российское или свободное прикладное программное обеспечение (ПО).

Анализ различных источников [1, 2, 3, 4, 5] привёл к формированию следующего списка пакетов для применения в указанных задачах.

- Caneda;
- Electric;
- Fritzing;
- gEDA;
- gLogic;
- Horizon EDA;
- KiCAD;
- KTechLab;
- Logisim (Logisim evolution);
- Oregano;
- QElectroTech;
- Qucs (Qucs-S);
- SimulIDE;
- WiRedPanda.

Из этого списка сразу следует исключить пакеты, в которых не поддерживаются функции моделирования: QElectroTech (обеспечивает только создание и редактирование принципиальных схем), Electric и Horizon EDA (реализуется только проектирование).

Данные по актуальности оставшихся пакетов и их наличию в российском открытом репозитории Sisyphus приведены в таблице 1

Таблица 1: Актуальность пакетов моделирования для электроники

Пакет	Дата первой версия	Текущая версия	Дата текущей версии	Версия в Sisyphus
Caneda	08.02.2016	0.3.0	12.06.2016	нет
Fritzing	14.08.2007	1.0.2	02.01.2024	0.9.6
gEDA	01.04.1998	1.10.2	22.12.2020	20060123 (23.01.2006)
gLogic	15.02.2012	2.6	01.11.2012	нет
KiCAD	1992	8.0.2	28.04.2024	8.0.2
KTechLab	28.01.2009	0.51.0	27.03.2023	0.51.0
Logisim	04.2001	2.7.1	21.03.2011	2.7.1
Logisim-evolution	31.07.2015	3.8.0	02.10.2022	нет
Oregano	1999	0.84.43	23.01.2020	0.84.43
Qucs	08.12.2003	0.0.19	22.01.2017	0.0.19
Qucs-S	26.01.2017	24.2.1	31.03.2024	24.2.1
SimulIDE	03.04.2021	1.1.0	25.03.2024	1.0.0
WiRedPanda	27.02.2016	4.1.12	08.06.202	нет

Для задач моделирования аналоговых схем наиболее простым решением служит **SimulIDE**. Недостатками пакета являются отсутствие поддержки языка **SPICE** и некоторых элементов схем (трансформатора, кварцевого генератора, предохранителя).

Наиболее сложными (профессиональными) вариантами являются **gEDA** и **KiCAD**, при этом моделирование в **gEDA** реализовано только с использованием командной строки.

KTechLab и **Qucs-S** по сложности применения при моделировании аналоговых схем занимают промежуточное положение, но в **KTechLab** отсутствует поддержка языка **SPICE**.

В докладе обсуждаются и другие параметры пакетов, потенциально пригодных для моделирования аналоговых схем.

Для задач моделирования цифровых (логических) схем наиболее простым решением является **WiRedPanda**, однако этот пакет имеет крайне ограниченный набор компонентов и может применяться только для самых простых задач. Также достаточно простыми в использовании являются **Logisim-evolution** и **SimulIDE**, при этом **Logisim-evolution** позволяет получить таблицу истинности для ло-

гической схемы, но не позволяет получить диаграммы уровней сигналов, а SimulIDE — наборот.

Наиболее сложными в применении при моделировании цифровых (логических) схем опять являются gEDA и KiCAD.

При моделировании цифровых схем важной возможностью является создание подсхем (блоков), которые затем используются как элементы более сложных схем. Эта возможность реализована в Logisim-evolution, SimulIDE, gEDA и KiCAD.

В докладе обсуждаются и другие параметры пакетов, потенциально пригодных для моделирования цифровых схем.

Исходя из функциональных возможностей и контекста использования возможны следующие варианты замены зарубежного ПО пакетами свободного ПО.

- KiCAD вместо Accel EDA, Altium designer, AutoCAD Electrical, AWR Design Environment, Circuit Maker и PCAD;
- Logisim-evolution вместо Intel Quartus (в части цифрового моделирования);
- Qucs-S вместо MicroCap, симуляторов SPICE (Pspice, LTspice) и в части задач вместо NI Multisim;
- SimulIDE вместо Atmel Studio и в части задач вместо NI Multisim.

Литература

- [1] CAD-программы: [Электронный ресурс]. URL: https://cxem.net/software/soft_CAD.php
- [2] Бесплатные онлайн симуляторы электрических схем для радиолюбителя и новичка: [Электронный ресурс]. URL: <https://dzen.ru/a/XpRbGHWqkVLR1A>
- [3] Программы моделирования электронных устройств на уровне принципиальных схем: [Электронный ресурс]. URL: http://library.voenmeh.ru/jirbis2/files/materials/ifour/book1/book_on_main_page/2.2.htm
- [4] Квалтырева Е. В., Минкин А. В. Анализ программ моделирования электрических схем и цепей для использования в учебных целях. / Форум молодых учёных. 2018. № 11(27). С. 846–848.
- [5] <https://packages.debian.org/stable/electronics/>

Елена Татьянич

Волгоград, ФБГОУ ВО «Волгоградский государственный социально педагогический университет»

Проект: Курс «3D-моделирование и 3D-печать»

http://lms.vspu.ru/courses/3d-modelling_and_3d-printing/

Использование свободного программного обеспечения при подготовке будущих учителей информатики в области 3D-печати

Аннотация

Доклад посвящён рассмотрению свободного программного обеспечения, позволяющего осуществить полный цикл создания объекта средствами технологии трёхмерной печати FDM: формирование трёхмерной модели (Blender, FreeCAD), подготовка её к печати (Blender, FreeCAD, Ultimaker Cura), генерация управляющего кода для 3D-принтера (Ultimaker Cura). Описан опыт использования указанного свободного ПО в рамках дисциплины «3D-моделирование и печать», реализуемой для бакалавров направления «Педагогическое образование» профилей «Математика» и «Информатика», «Информатика» и «Технология».

Ключевые слова: *3D-печать, редактор 3D-графики, САПР, слайсер, подготовка будущих учителей информатики.*

Подготовка будущих учителей информатики в области преподавания технологий трёхмерной печати включает два направления:

1. Подготовку специалистов в области трёхмерной печати.
2. Подготовку педагогов, владеющих опытом реализации образовательных программ, предполагающих изучение и использование технологий 3D-печати. Исследование этого направления остаётся за рамками доклада.

Осуществление первого направления подразумевает овладение опытом реализации полного цикла создания объекта средствами технологий 3D-печати¹, что включает [1]:

¹Изучение технологий 3D-печати в школе осуществляется на примере технологии FDM или её свободного аналога FFF как наиболее простых в освоении и использовании, безопасных, обеспеченных широким выбором дешёвых и экологических расходных материалов.

1. подготовку трёхмерной модели;
2. конвертацию её в формат STL;
3. дополнительную обработку конвертированной модели и её передачу на 3D-принтер;
4. настройку принтера;
5. печать модели;
6. постобработку распечатки.

Рассмотрим возможности СПО для обеспечения реализации первых трёх этапов рассмотренного цикла. На остальных ПО отвечает за работу конкретного принтера (прошивка, этапы 4,5) или не используется совсем (6). Выбор рассматриваемых далее представителей СПО обусловлен или отсутствием выбора в классе соответствующего СПО (САПР **FreeCAD**), или богатым профессиональным инструментарием и обширным удобно структурированным разделом справочных материалов сайта разработчика (пакет создания 3D-графики **Blender**), или дружелюбностью для начинающих и широтой возможностей (слайсер **Ultimaker Cura**).

1) В зависимости от целей дальнейшего использования распечатки создание и первичная подготовка 3D-модели может осуществляться или в **Blender**, или в **FreeCAD**. Обе среды позволяют:

- создавать твердотельные модели;
- проверять их манифолдность и направленность нормалей граней наружу модели;
- оценивать размеры модели относительно печатной области 3D-принтера;
- оптимизировать печать путём использования в модели фасок;
- оценивать необходимость наличия мелких и/или нависающих элементов модели;
- экспортировать модель в формат STL.

Обе системы предоставляют возможность создавать модели из примитивов с использованием булевских операторов (объединение, пересечение, разность), операций выдавливания, масштабирования, поворота, отзеркаливания. Но применение **Blender** наиболее оправдано для создания объектов художественной направленности. Построение по точным размерам в нём затруднительно, хотя и возможно. Для создания моделей инженерной направленности предназначен

FreeCAD, ориентированный на контроль за точностью размеров моделируемых объектов и историей их моделирования через дерево проектов. **FreeCAD** так же поддерживает построение моделей по чертежам, использование системы автоматических ограничений параметров чертежа, выдавливание модели по чертежу.

В **Blender** представлены встроенные инструменты проверки на манифолдность и плагин **3D Print Toolbox**, позволяющий автоматически исправлять найденные проблемы. Аналогичные инструменты в **FreeCAD** собраны в верстаке **Part** (Проверка геометрии).

2) Конвертация модели в формат **STL** поддерживается всеми средами создания **3D**-моделей.

3) **Ultimaker Cura** позволяет выбрать марку используемого принтера. В случае отсутствия нужного принтера в списке минимальные параметры печати (размер области печати, количество экструдеров, диаметр сопла) можно задать вручную. **Cura** предоставляет широкий выбор сортов пластика с предоставленными фирмами-производителями сведениями (температурой плавления, диаметром прутка и т. п.), которые автоматически учитываются в настройках параметров печати. **Cura** предлагает выбор из предустановленных стандартных настроек печати, подобранных для ускорения печати, повышения её качества или баланса качества и скорости. Все настройки подлежат коррекции.

Cura позволяет регулировать высоту слоёв (начального, верхнего, обычных), стенок, заполнение (тип, плотность в процентах), скорость печати (скорость, ускорение рывок), скорость подачи пластика, температуру, обдув, ретракт, использование средств адгезии (отсутствие, рафт, юбка, кайма), наличие поддержек, их расположение и вид. **Cura** автоматически исправляет некоторые геометрические ошибки создания модели.

Также **Cura** позволяет изменять пропорции модели, её положение на печатном столе (переносы, повороты). Последнее позволяет ускорить печать, убрав необходимость использования поддержек, или предотвратить некоторые ошибки печати.

После нарезки на слои (фактически, генерации управляющего кода для принтера) **Cura** оценивает время печати и объём требуемого пластика.

С 2020 года **Blender**, **FreeCAD**, **Ultimaker Cura** обеспечивают реализацию курса «3D-моделирование и печать» (направление «Педагогическое образование» профили «Математика» и «Информатика»,

«Информатика» и «Технология», ВГСПУ). Курс включает разделы «3D-моделирование», «3D-печать». Содержит 9 лекций, 14 лабораторных и 2 контрольные работы, задания для СРС. Зачёт — защита индивидуальных проектов. Курс поддержан полным комплектом методических материалов.

Литература

- [1] *Татьянич Е. В.*, Структура и модели формирования готовности будущего учителя информатики к обучению технологиям трёхмерной печати учащихся школ // Мир науки. Педагогика и психология, 2021 №2, <https://mir-nauki.com/PDF/17PDMN221.pdf> (доступ свободный). Загл. с экрана. Яз. рус., англ.

Дмитрий Литвинов

Москва, ГКУ «Инфогород», департамент решений в сфере образования, отдел проектирования интерфейсов и дизайна управления функционального развития программы «Московская электронная школа»

СПО и проектирование интерфейса пользователя

Аннотация

В данной статье рассматривается проблема острой нехватки программных продуктов с открытым исходным кодом для работы в сфере дизайна и, в частности, проектирования интерфейсов.

Ключевые слова: *UI/UX, дизайн интерфейса.*

Рассмотрим модель разработки программного обеспечения, далее просто ПО, под названием «Водопад», представленную на Рис. 1 [1]. Эта схема является наиболее подходящей для описания проблемы данной темы. Возникшие проблемы и задержки в одном из первых трёх этапов будут иметь большие негативные последствия в последующих этапах, и, как следствие, для продукта в целом.

Итак, мы видим пять этапов разработки ПО, по крайней мере два из которых не имеют очевидной зависимости от использования свободного программного обеспечения (СПО). Это этап номер один «Проектирование» и этап номер пять «Поддержка». В целом к ним можно добавить и четвёртый этап, но технологий тестирования много, и некоторые могут включать в себя тестирование с применением различного ПО.

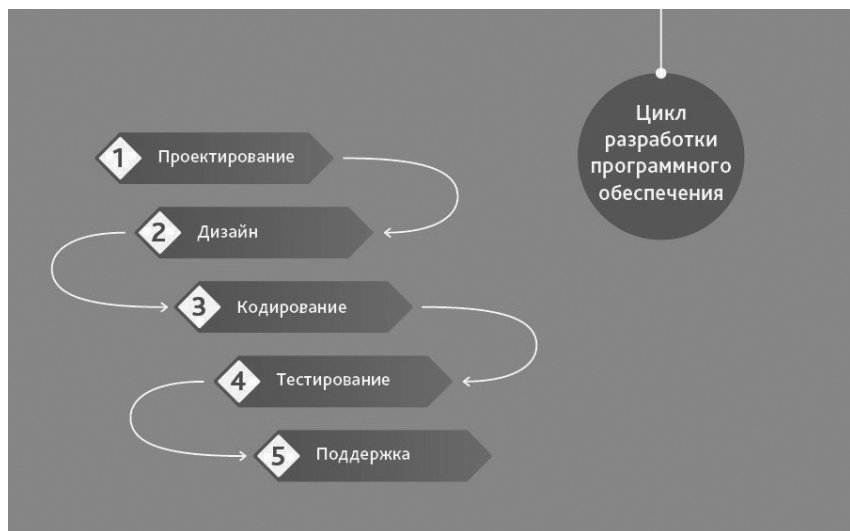


Рис. 1: Цикл разработки программного обеспечения.

Намного сильнее зависят от типа лицензии ПО оставшиеся два этапа: это «Дизайн» и «Кодирование».

И если у программистов при переходе на СПО и Linux имеется обширный инструментарий, то дизайнеры сталкиваются с почти полным отсутствием альтернатив ПО в данной сфере деятельности. Таким образом, пока также отсутствует возможность внедрения СПО для проектирования и дизайна в образовательный процесс в школах, колледжах и вузах.

Ниже приведена таблица программных продуктов, которые могут быть применены для проектирования интерфейсов пользователя на разных стадиях развития продукта — от проектирования интерфейса до презентации готового продукта заказчику. Данные программы имеют, в той или иной степени, большие сообщества пользователей, а значит, достаточный объём учебного материала.

Если убрать из рассмотрения несвободное и ПО под санкциями, а также ПО, которое не создано изначально для дизайна интерфейсов, из рассмотренных продуктов с полным циклом проектирования интерфейса останется только Penpot. Это проект с небольшим со-

Наименование	Входит в «промышленный стандарт»	СПО	Сфера применения
Gimp	Нет	Да	Фото, цифровая живопись
Krita	Нет (но уже близок)	Да	Цифровая живопись
Inkscape	Нет	Да	Иллюстрации
Figma	Да	Нет	UI\UX дизайн, прототипирование
Lunacy	Да	Нет	UI\UX дизайн, прототипирование
Penpot	Нет (в перспективе войдет)	Да	UI\UX дизайн, прототипирование
Quant UX	Нет	Да	UI\UX прототипирование
Wireflow	Нет	Да	UI\UX прототипирование
Scribus	Нет	Да	Типография
Blender	Да	Да	3D графика, моушен-графика

Рис. 2: Сравнительный анализ СПО для дизайна и проектирования интерфейсов.

обществом пользователей и таким же малым количеством учебного материала, который всё ещё находится в разработке. Но так как он является аналогом, в хорошем смысле этого слова, его проприетарным конкурентам в лице **Figma**, **Sketch** и **Lunacy**, то переход на него может пройти достаточно гладко. Автору статьи потребовалось не более пятнадцати минут, чтобы освоиться с функционалом данного приложения.

Но есть и ряд проблем. Во первых, это веб-приложение, которое, как и его веб-конкуренты, имеет существенные минусы:

1. привязка к наличию и качеству интернет-соединения;
2. техническое ограничение одной вкладки браузера, в среднем, в 2 гигабайта оперативной памяти [2];
3. собственно сам браузер, требующий и потребляющий немалое количество ресурсов «железа» в отличие от нативных решений.

В теории, эти минусы обходятся установкой **Penpot** в собственный контейнер [3]. На сайте проекта есть соответствующая инструкция,

но маловероятно, что это будет производиться в образовательных организациях.

Второе ограничение — разделение проекта на мелкие подпроекты и библиотеки. А третье — это необходимость обновления парка рабочих машин.

Есть ещё один минус, который невозможно обойти — проект является иностранным продуктом, и пока непонятно, как его руководство будет вести себя по отношению к коммитам от наших соотечественников.

В итоге, мы имеем практически полное отсутствие программного обеспечения для проектирования и прототипирования интерфейсов на базе СПО. И для решения этой проблемы сообществу российского СПО необходимо консолидироваться. Видится несколько путей решения отсутствия ПО для разработки интерфейсов:

1. присоединение к разработке уже имеющихся инструментов;
2. создание форков данных проектов, что позволит не зависеть от иностранного влияния;
3. создание с нуля нативных решений в рамках поддержки российских компаний государством.

Литература

- [1] *Alva, Thomas*, Ещё раз про семь основных методологий разработки, 2015, <https://habr.com/ru/companies/edison/articles/269789/>
- [2] *Figma Learn*, Reduce memory usage in files, 2024, <https://help.figma.com/hc/en-us/articles/360040528173-Reduce-memory-usage-in-files>
- [3] *PenPot*, Self-host Penpot, 2024, <https://penpot.app/self-host>

Михаил Тергоев, Александр Давыдов, Глеб Орлов, Георгий Янковский

Санкт-Петербург, ООО «Базальт СПО», ФГБОУ ВО «БГТУ «ВОЕНМЕХ» им. Д. Ф. Устинова»

Проект: Ingame <https://github.com/Castro-Fidel/ingame>

Разработка программы с графическим интерфейсом для запуска игр с помощью геймпада

Аннотация

Сфера компьютерных игр является одной из быстроразвивающихся индустрий в мире, в том числе приобретает популярность на платформах GNU/Linux. Ввиду растущего спроса на связанное с играми СПО необходимо разработать программный продукт, позволяющий запускать видеоигры для разных платформ из одного меню. В докладе приводится описание этапов разработки приложения с графическим интерфейсом запуска установленных через PortProton игр и программ с помощью геймпада. Продемонстрированы результаты использования данного меню при управлении игровым контроллером.

Ключевые слова: *видеоигры, PortProton, python, QT, QML.*

На данный момент индустрия компьютерных игр является одной из быстроразвивающихся индустрий в мире, интерес к которой в пределах России остаётся высоким. Большинство видеоигр выпускается в виде исполняемых программ для зарубежной проприетарной ОС Microsoft Windows (как правило, для архитектур CPU x86 и AMD64 в виде 16-, 32- и 64-битных приложений). Сообществом свободного программного обеспечения разрабатывается проект Wine, который формирует слой совместимости для запуска бинарных файлов с расширением «.exe» на системах GNU/Linux, таким образом позволяя работать с ПО, созданным исключительно для ОС семейства Windows [1].

В 2018 году компания Valve приняла решение сделать своё отвлечение «Valve Proton» от вышеописанного проекта и реализовала модификацию для улучшения совместимости именно с видеоиграми, которая увеличивает производительность многопоточных игр, обеспечивает трансляцию вызовов DirectX 11 и 12 через API Vulkan, улучшает совместимость с игровыми контроллерами [2].

Существует отечественная утилита под названием PortProton, предоставляющая пользователю графический интерфейс для автома-

тической установки видеоигр (например, OSU, Genshin Impact), игровых центров (например, Lesta Game Center, vkPlay Games Center), эмуляторов платформ электронных приставок (например, эмулятор платформы игровой приставки Sony PlayStation Portable — PPSSPP), управления Wine-префиксами, запуска сторонних Windows-совместимых приложений и расширенной настройки [3].

Ввиду отсутствия у вышеописанной утилиты возможностей выбора видеоигр для запуска с помощью геймпадов и просмотра списка нативных для GNU/Linux приложений из категории «Игры» из репозитория системы нашей командой было принято решение о разработке проекта «Ingame», решающего вышеописанные задачи.

Для формирования отзывчивого и удобного меню можно воспользоваться технологией Qt 6 QML, которая позволяет реализовывать графический интерфейс с использованием аппаратного ускорения через слой OpenGL. В качестве языка для программирования части обработки бизнес-логики был выбран Python 3.x ввиду популярности среди открытого программного обеспечения. Для связки Python с Qt (в том числе — QML) используется прослойка PySide. Проект Ingame использует менеджер зависимостей Poetry.

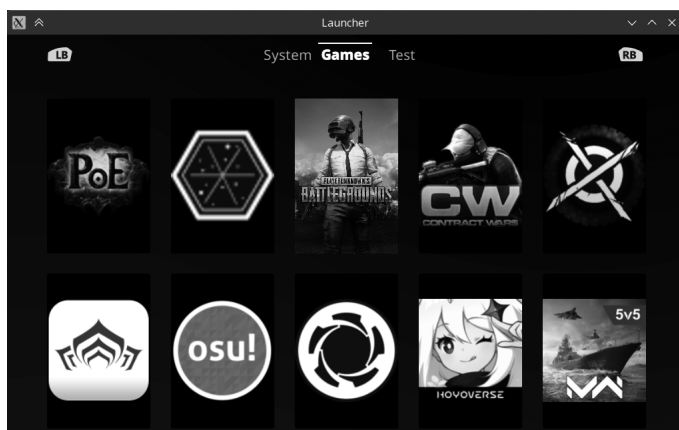


Рис. 1: Главное меню запуска игр через PortProton.

Список сторонних игр, запускаемых через Wine, запрашивается из сформированной утилитой PortProton директории. Информация

о каждой выбранной игре загружается с ресурса цифрового распространения видеоигр **Steam**. Меню программы масштабируется под размер экрана. Управление реализовано как с помощью контроллера для Xbox 360 (стрелки на левом верхнем стике — выделение элементов в сетке, А — выбрать выделенный элемент, В — назад), так и при использовании клавиатуры (Tab — переключение фокуса на соседний правый элемент, Shift+Tab — фокус в обратную сторону, то есть на соседний левый элемент, Пробел — выбрать выделенный элемент, Esc — назад). Помимо геймпадов, поддерживается управление с сенсорного экрана.

Программа имеет несколько вкладок: управление питанием используемого устройства (например, выключить, перезагрузить), список игр **PortProton**, установленные нативные видеоигры, системные настройки (например, произвести переход в полноэкранный режим). После выбора конкретного наименования происходит переход в меню с информацией и кнопками запуска игры. На рисунке 1 продемонстрирован список игр, показываемый после запуска приложения.

Проект «Ingame» размещён на хостинге IT-проектов Github, имеет лицензию MIT, все желающие могут присоединиться к разработке.

Литература

- [1] WineHQ — Run Windows applications on Linux, BSD, Solaris and macOS, <https://www.winehq.org/>
- [2] Introducing a new version of Steam Play, <https://steamcommunity.com/games/221410/announcements/detail/1696055855739350561>
- [3] PortProton — ALT Linux Wiki, <https://www.altlinux.org/PortProton>

Екатерина Бесчервтная, Анастасия Волкова, Владислав Агабекян, Владимир Симонов

Москва, ФГБОУ ВО «Российский государственный социальный университет»

Проект: Информационная система управления учебным макетом летательного аппарата <https://github.com/EkaterinaBes/IMUVizual>

Симуляция пилотирования летательного аппарата с помощью информационной системы управления учебным макетом

Аннотация

Представлена разработанная авторами информационно-измерительная система, предназначенная для пилотирования учебным макетом летательного аппарата. Использование системы обусловлено необходимостью получения студентами навыков разработки модулей учебного макета летательного аппарата. В разработке использован модуль инерциальной навигации IMU-9 DOF, иные датчики информации, а также программные средства визуализации данных. Приведены результаты экспериментов.

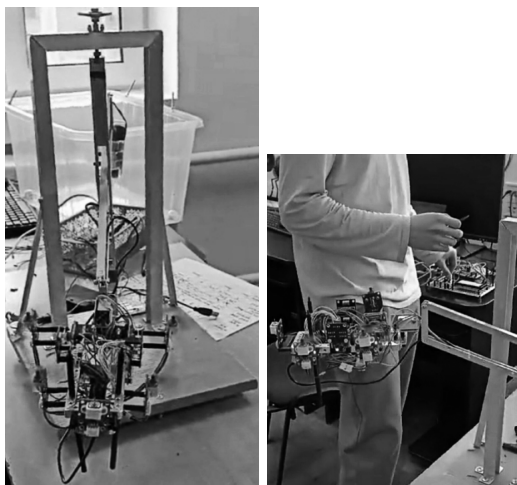
Ключевые слова: *информационная система управления, летательные аппараты, пилотирование летательного аппарата, учебный макет, модуль IMU-9 DOF.*

Тенденция перехода с пилотируемых летательных аппаратов на беспилотные ставит перед разработчиками данных систем новые задачи. Важно готовить специалистов для пилотирования БПЛА. Для выполнения этого требования нужна разработка система управления, которая поможет в рамках учебной деятельности тренировать пользователей, совершенствовать их знания и навыки пилотирования.

В статье рассматривается информационная система управления учебным макетом летательного аппарата на базе платформы Arduino. Для программной разработки применены такие среды, как Arduino IDE и Processing 3.

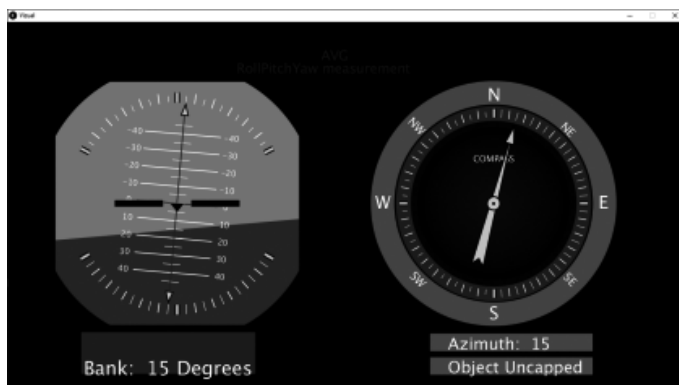
В ходе работы были проведены эксперименты: запущена система, в которой предусмотрены два режима функционирования: ручное и автоматическое управление. Эти варианты работы системы выбираются и применяются пользователем на наземной части — цепи, представляющей собой управляющие компоненты, подключённые к

микроконтроллеру Arduino Mega 2560. Управляющими компонентами названы два ползунковых потенциометра Trema-SLIDERS и энкодер. Сам макет летательного аппарата закреплён на весах Роберваля и является подвижной частью системы. На другом плече находится противовес. В подвижной части системы есть модуль IMU-9 DOF. Он считывает углы Эйлера. Эти углы влияют на крен, тангаж и курс летательной машины в полёте. Значения углов задаются управляющими элементами наземной части системы. Полёт летательного аппарата возможен благодаря пяти двигателям с пропеллерами. К наземной и подвижной частям системы подключено по одному радиомодулю для беспроводной передачи данных. Ручной режим управления заключается в мониторинге пользователем полёта и адаптации частоты вращения пропеллеров в соответствии с требованиями. Так, за счёт сдвига ручки ползунков и поворота ручки энкодера пользователь добивается взлёта, посадки, поворота и крена летательного аппарата. Чем больше мощность подаётся на двигатели, тем выше поднимается летательная машина.



Автоматический режим управления заключается в том, что пользователь добивается набора высоты летательного аппарата с помощью одного ползунка: при сдвиге его ручки задаётся желаемая высота полёта, и далее происходит стабилизация частоты вращения моторов

и, как следствие, текущей высоты. Данные о высоте передаются с закреплённого на летательном аппарате дальномера HC-SR04.



Главную роль в визуализации играют программный код на Processing, при запуске которого на мониторе отображаются бортовые приборы, — авиагоризонт и компас, FPV-очки с подключёнными по радио каналу к ним FPV камерой и символьный ЖК-дисплей. FPV-камера фиксирует то, что видит перед собой. Перед ней может находиться плакат с горизонтом, и тогда пилот в очках будет погружён в симуляцию так, как если бы являлся пилотом самолёта. ЖК-дисплей показывает режим, углы Эйлера, желаемую и текущую высоту.

Были проведены экспериментальные запуски. Результаты представлены на рисунках.

Литература

- [1] Скворцова М. А. БПЛА вертикального взлёта и посадки для мониторинга местности, <https://www.elibrary.ru/item.asp?edn=gvvzso>
- [2] Симонов В. Л., Ерпелев А. В., Давыдова Е. К., Хохлов Е. Г. Моделирование системы управления вертикальным взлётом и посадкой, <https://www.elibrary.ru/item.asp?edn=edvgt0>
- [3] Венрева Е. Л., Махонина А. Н. Разработка систем управления моделями наземных и воздушных беспилотных транспортных средств, <https://www.elibrary.ru/item.asp?edn=kjnppe>
- [4] Программа Arduino IDE, <https://all-arduino.ru/arduino-ide/>.

Милена Коул

Москва, ГАУ «Центр цифровизации образования»

Проект: OpenBoard <https://github.com/OpenBoard-org/OpenBoard>

Доработка СПО по запросам образовательных учреждений

Аннотация

В образовательных учреждениях могут возникать уникальные для каждой конкретной организации требования к ПО. Свободное ПО позволяет вносить доработки самостоятельно, не имея зависимости от других его разработчиков, что очень удобно в подобных случаях. Различные образовательные учреждения имеют возможность дорабатывать ПО под свои нужды, не затрагивая при этом остальных пользователей данного ПО. Это можно видеть на примере OpenBoard — свободное программное обеспечение, которое используется в образовании в различных странах. Образовательные системы разных стран отличаются друг от друга, поэтому не все требования к доработкам, которые возникают в России, нужны в других регионах.

Ключевые слова: *OpenBoard, апстрим, патчи, ответвление в разработке.*

В настоящее время во многих школах на уроках используются электронные доски. Работу на них сложно представить без OpenBoard — программы, позволяющей имитировать работу на обычной меловой доске [1]. В ней присутствует множество различных инструментов [2], которые заменяют инструменты для работы с доской из жизни, а также дополнительные виджеты для проведения интерактивных уроков.

В Центре цифровизации образования при Правительстве Москвы в настоящее время разрабатывается операционная система МОС, которая уже внедрена в большинстве московских школ. Также разработчики МОС дорабатывают приложения, основываясь на запросах, которые поступают от самих учителей, в том числе и по OpenBoard.

За последние два года были реализованы такие возможности этой программы как векторизация иконок, сохранение размера сетки, локализация, возможность задавать цвет фона, корректное отображение в системе с тёмной темой и многие другие, в том числе описанные в данной статье.

Линии и векторы

Одна из первых задач, которая поступила от московских учителей — добавление стилей линии, а также добавление нового инструмента — вектора. Изначально в **OpenBoard** был инструмент линия, который позволял рисовать только сплошную линию. Но, как правило, в школах на различных уроках используются не только сплошные линии, но ещё и пунктирные. Так были добавлены ещё два стиля линий:

- Пунктир;
- Пунктир точками.

Также для упрощения работы на уроках математики был добавлен новый инструмент — вектор. Это, по сути, та же линия, но только со стрелкой на конце, указывающей направление. Для вектора также было сделано три стиля с различным направлением стрелок. Задача была выполнена и отправлена в апстрим. Но через какое-то время последовал ответ от разработчиков **OpenBoard**, что в данном функционале они не нуждаются, так как стараются сохранить минимализм в **OpenBoard**.

При этом периодически в апстриме появляются запросы от других учителей на добавление стилей линии. Разработчики апстрима не принимают участия в этих обсуждениях, но пользователи программы могут применять патч, реализованный в команде МОС.

Циркуль

Также в **OpenBoard** есть виджеты, которые имитируют настоящие инструменты, например, циркуль. При реализации определённых возможностей цифровой аналог данного инструмента может оказаться намного удобнее в использовании. Таким образом, от пользователей поступил запрос на улучшение циркуля. Было добавлено несколько новых возможностей:

- Пунктир;
- прилипание циркуля к сетке;
- возможность задавать точный градус поворота циркуля (с отрисовкой и без);
- фиксированный поворот на 90 градусов против часовой стрелки.

Данные изменения также были предложены апстриму. Они отказались их принимать, так как сами занимаются доработкой данного

инструмента. Как выяснилось, учителям требовался радиус меньше, чем могла позволить реализация инструмента из апстрима. С учётом поставленной проблемы, автором статьи был видоизменён циркуль, который теперь может рисовать круги со столь маленьким диаметром, что этот круг практически является пикселем.

Multi-touch

Помимо всего этого, на протяжении нескольких последних лет в апстрим поступал запрос на новый функционал — **multi-touch**. Разработчики сразу сообщили, что реализовывать его не планируют, так как данный функционал не является необходимым для их пользователей. При этом во многих школах, в том числе и российских, данная возможность была необходима для одновременной работы у доски нескольких учеников. Автору статьи удалось реализовать **multi-touch**, и в московских школах уже стоит сборка **OpenBoard** с этим функционалом.

Несмотря на сложности коммуникации с апстримом, разработчики МОС производят поддержку сборки **OpenBoard** со всеми доработками для российских учителей [3], чтобы работа в этой программе была удобной и комфортной. Пакет размещён в репозитории ROSA Linux, общем с МОС.

Литература

- [1] OpenBoard, 2024, <https://github.com/OpenBoard-org/OpenBoard/blob/master/README.md>
- [2] OpenBoard Tutorial, 2022, https://openboard.ch/download/Tutoriel_OpenBoard_1.6EM.pdf
- [3] Режим доски (OpenBoard), 2024, <https://school.mos.ru/help/instructions/interactive-panel/programs-panel/openboard/>

Андрей Пименов

Санкт-Петербург, Университет ИТМО

Проект: VectorNST, VectorWeaver, CoverGAN, EvoVec, VectorMem

<https://github.com/CTLab-ITMO/VGLib>

Открытая библиотека методов анализа и генерации векторной графики

Аннотация

Современные подходы компьютерного зрения по генерации и обработке изображений посвящены в основном растровой графике. Растровые изображения представляют собой многомерную матрицу пикселей, однако существует и другой способ описания изображений — векторный подход, использующий математические формулы для описания объектов. К сожалению, методы обработки векторной графики в настоящее время изучены недостаточно, и не существует единой библиотеки с понятным интерфейсом, которая бы позволяла производить различные операции с векторными изображениями. Предлагаемая нами открытая библиотека позволяет закрыть этот пробел и объединить ранее предложенные подходы по генерации, анализу и обработке векторных изображений. Библиотека включает в себя методы дифференцируемой растеризации, векторизации, нахождении плагиата векторных чертежей, генерации векторных изображений по музыке и переносу стиля между векторными изображениями. В дальнейшем планируется улучшать библиотеку и добавлять в неё новые методы генеративной обработки векторной графики.

Ключевые слова: *векторная графика, нейронные сети, трансформеры.*

EvoVec

На данный момент вопрос векторизации полностью не решён. Существующие подходы имеют ряд недостатков как в качестве, так и скорости работы. Для решения этих проблем предлагается эволюционный алгоритм векторизации изображений. Основная идея подхода заключается в итеративном улучшении векторных изображений с помощью мутаций и кроссоверов. Сам алгоритм не требует никаких параметров, таких как количество путей в векторизованном изображении, кроме самого исходного изображения, что делает алгоритм универсальным. Результаты сравнения с существующими решениями

показывают, что разработанный алгоритм качественно и быстро векторизует изображения, превосходит другие по попиксельному MSE на 15% и не генерирует ненужных путей в итоговом векторизованном изображении.

VectorNST

Наше исследование показывает, что применение стандартных функций потерь стиля и контента меняют стиль рисования векторных изображений, поскольку структура векторных примитивов отличается от пикселей. Чтобы справиться с этим, мы вводим новые функции потерь. Предлагаемое нами решение, **VectorNST**, основанное на дифференцируемой растеризации, использует эти новые функции потерь и может изменять параметры цвета и формы изображения содержимого так, чтобы соответствовать изображению стиля. Качественные эксперименты демонстрируют эффективность предложенного метода **VectorNST** по сравнению с современными подходами переноса стиля для растровых изображений и единственным существующим подходом для стилизации векторных изображений **DiffVG**.

VectorWeaver

Диффузионные растровые модели не могут успешно справляться с генерацией векторных изображений из-за существенных различий в структуре изображений. Мы решаем проблему синтеза векторных изображений, разрабатывая новую архитектуру модели на основе диффузии и трансформеров, которую мы назвали **VectorWeaver**. Для обучения модели мы собрали набор данных векторных изображений из общедоступных ресурсов и провели их аугментацию. Качественные эксперименты демонстрируют превосходство и вычислительную эффективность предложенной модели по сравнению с существующими методами генерации векторных изображений.

CoverGAN

Основной архитектурой модели для генерации векторных обложек была выбрана условная генеративно-состязательная сеть. Она обучается на датасете из собранных ранее растровых обложек музыкальных композиций. В качестве условия сеть принимает саму музыкаль-

ную композицию, из которой затем выделяются разные музыкальные признаки, а также эмоцию, которую музыка вызывает у слушателя. Задача генератора состоит в том, чтобы на основе этих данных выдать векторную обложку, максимально соответствующую условию. Для трансформации переданных данных в конечный результат планируется использовать алгоритм **Path Decoder**, описанный в статье **Im2Vec**. Также реализована модель, добавляющая надпись, содержащую информацию об исполнителе и названии трека, на обложку. Таким образом, сервис создаёт для музыкальной композиции обложку в векторном формате, отвечающую эмоции, заданной пользователем.

VectorMem

Одной из проблем анализа векторной графики является неограниченность векторных изображений в количестве объектов на изображении. Это накладывает существенные ограничения в плане анализа и генерации новых изображений. Мы хотим предложить решение, которое поможет обрабатывать векторные изображения неограниченной сложности. Для этого мы разработали модель **VectorMem** в которой использовали наработки в области трансформеров памяти. Для проверки её работы мы протестировали её на задаче поиска плагиата на одних из самых высокоразмерных векторных данных — чертежах. Результаты показали её превосходство как над растровыми моделями, так и на обычных решениях анализа векторной графики.

Стас Фомин

Москва, ИСПРАН

Проект: **PyAlgovizualizer** github.com/belonesox/pyalgovisualizer

PyAlgovizualizer — эффективное преподавание алгоритмов

Аннотация

Преподавание алгоритмов через книги-слайды — устарело и неэффективно. С другой стороны, путь постижения хакера — реальная отладка на промышленных языках — тяжела и не для всех.

Оптимальный вариант — максимально компактный и читаемый алгоритм, который можно выполнять, отлаживать, исследовать, и при

этом — магически визуализируется то, что происходит внутри — структуры данных и т.п. Причём чтобы обучающемуся ничего не нужно было настраивать.

`PyAlgoVizualizer` — это как раз такой модуль и подход, как готовить «живые» самовизуализирующиеся, компактные алгоритмы на Python, по которым преподаватель может провести занятие вживую, или записать видеоролик, и которых студенты могут вполне изучать самостоятельно, имея только браузер.

Ключевые слова: *преподавание алгоритмов, PyAlgoVizualizer, Python.*

Тезисы

По личному опыту — 25 лет преподавания (МФТИ, ИСПРАН), и 35 лет учеником (ВУЗЫ, курсы, МООСи), преподавание алгоритмов, как у нас, так и по миру, сильно не дотягивает до возможного технологического уровня.

Да, когда-то, в ветхозаветные времена — времён поклонению библиям алгоритмов — трёхтомнику Кнута, «книжке с ослом» («Алгоритмы: построение и анализ»), которых правда обычно держали на полках, а реально читали плохонапечатанные методички ВУЗов, казалось — что так и должно быть, хотя работало это только для тех, кто реально пытался всё это как-то кодить.

Но надо признать очевидные «для 2024» факты: «бумажные книги — всё».

По форме:

- нельзя искать,
- неудобно (для 2024) читать,
- невозможно копировать и как-то взаимодействовать.
- кирпич! не унести, не скопировать, не ...
- качественная печать — белая бумага, цветные иллюстрации — адски дорого, обычно на бумаге «стандарт Питер-пресс» с серой «туалетной бумагой».

Любители «запаха бумажных листов», «шелеста страниц», «разрезания упаковки», «читать про алгоритмы в метро» — мною не наблюдались уже давно.

Также бумага неудобна по процессу издания:

- «вотерфолл»-процесс, с фиксированным дедлайном и невозможностью правок.

- куча дополнительных печатных ограничений хорошей вёрстки (совпадение строчек на прямой и обратной стороне листа и т.п.), которой правда уже много кто не заморачивается.

Давайте не тратить кучу сил на вёрстку, впихивание слабовпихуемого в прокрутово ложе «листа бумаги», «чернобелой печати», когда это просто ненужно — читать будут с разных устройств, телефонов, планшетов, портретно и альбомно ориентированных мониторов разного разрешения, где нужна адаптивная вёрстка, нет ограничений на объём, но есть цвета и гиперссылки и возможность интерактивности, которые пропадают в книгах. Не нужны ни бумажные учебники, ни специализированные гибкие-негибкие e-ink устройства — это я говорю, как владелец нескольких десятков «электронных книг».

Так что проехали! Не говоря уж о том, что в РФ полезно бы вообще держаться подальше от издательского бизнеса учебников — там до сих пор мафия, перестрелки, бандиты с золотыми цепями (опыт личных знакомых «в теме»).

Казалось бы — электронные PDF-книги, прекрасно написанные, адаптированные для чтения с планшетов, с правильными экранными шрифтами, мониторной вёрсткой, с цветными картинками и гиперссылками, читаем на компьютерах и планшетах, публикуем-обновляем на сайте сами, используем цвет-поиск... — проблема решена?

Это действительно сильно лучше, но тут по-прежнему активны ограничения статики:

- страничная вёрстка — с кучей проблем автору по вёрстке с учётом разбиения, переносов, ссылок на изображения и картинки;
- проблема как показать работу алгоритма в динамике — серия иллюстраций, логи работы, и надежда, что где-то в голове читающего это соединится правильным образом. Мы так делали пару десятков лет, с автогенерацией иллюстраций-логов и читаемого алгоритма по реально работающему алгоритму, но даже у нас сложилось понимание ограниченности этого подхода.

Более того, галопирует инфляция статического контента — часто бесполезно и даже вредно писать книги — скорее всего по большей части темы — нужные книги уже написаны, получится или хуже, или толще, а с ростом толщины читаемости книги автоматически падает, а то, что вы хотите добавить, надо делать как-то максимально компактным и гибким, и дешёвым для разработки.

И главное — мир алгоритмов огромен, десятки тысяч статей ежегодно, непонятных, часто с кризисом воспроизводимости [1] — их нельзя засовывать в книгу. Наоборот, пришло время вытаскивать всё из книг, и превращать в некие «интерактивные лаборатории с нелинейной навигацией», как «структуры технологий» в популярных играх.

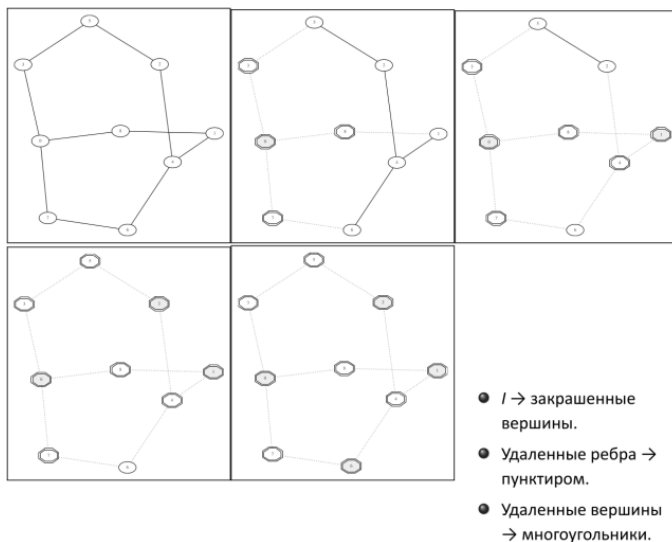
Сначала пройдемся по явно неудачным концептам, чтобы их избегать:

- «Алгоритм словами» — полный отстой. Может создать иллюзию понимания, но скорее всего ложную.
- «Псевдокод» — иллюзия понимания, но есть проблема воспроизводимости — при переходе к работающему алгоритму, может выясниться, даже при правильной реализации, что алгоритм вовсе не работал. Дилемма «Точность против Понятности» конечно очевидна, но псевдокод, каким бы понятным он был, за пределами правильного решения.
- Проблемы подачи контента — как говорилось выше, статика типа книг-статей-слайдов не передают динамики алгоритма — что-то можно делать с автогенерацией слайдов или картинок, но это очень мучительно и ограничено.

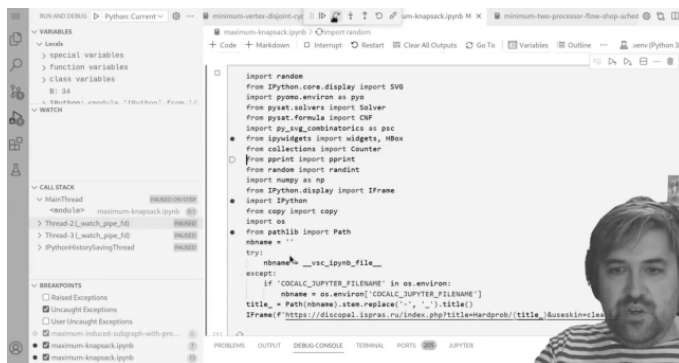
Это решения с более интересны, но тоже имеют свои недостатки:

- Процедурная видеоанимация (пакет `manim`) — красиво, но непонятно, что выполняется, сложно исследовать, нет синхронизации между роликами и выполняемым кодом.
- «Literate programming» — что-то подобное я делал ранее в книге и слайдах — генерация из питон кода, компактного описания алгоритма картинок и слайдов, но явно вижу пределы, и это сильно мучительно.

```
def greedyMIS(G)
    I = set([])
    while G.number_of_nodes() > 0: # цикл 0(n)
        n = list(G.nodes())[0]      # добавляем первую
        I.add(n)                    # попавшуюся вершину
        for s in list(G.neighbors(n)): # удаляем соседей
            G.remove_node(s)
        G.remove_node(n)            # и исходную вершину
    return I
```

- Jupyter-ноутбуки. Хороши для представления однонаправленных процессов (пайплайнов ML), постановок ЦЛП/SAT-задач, но не для самих алгоритмов, где происходят нетривиальные манипуляции — хотя в принципе, что-то достаточно неплохо можно показывать с поддержкой отладки в VSCode.



В идеале ожидается «живой» алгоритм, с которым можно играть-ся, изучать=экспериментировать, анализировать каждый шаг, но при

этом чтобы было понятно и быстро. «Talks is cheap, show me the code» © — казалось бы надо просто написать очень понятный код, в видеоролике разбирать его работу, и дать возможность легко обучающемуся повторить всё это, поэкспериментировать с каждой строчкой, покачать разные «что-если». Так в общем, уже и делает «весь интернетный мир», при эффективном изучении всего связанного с IT и это хорошо.

Но есть проблемы такого подхода, по отношению к преподаванию сложных алгоритмов:

- Алгоритмы на индустриальных языках (C/C++/Java/Rust...) — чудовищная перегрузка ненужными подробностями.
- проблема с визуализацией изменения состояния — просто тыкать в отладочные описания структур — неэффективно — чуть помогает расширение VSCode «Debug Visualizer» — но там долго перерисовывается, очень нетривиально описывать визуализацию (JSON-программирование).

Итак, наконец, представляем решение, которое мы используем несколько лет, при преподавании алгоритмов — от основ и тренировки решения «задач на собеседовании», до сложных алгоритмов комбинаторной оптимизации.

Решение — разработанный питон-модуль `pyalgotvisualizer` [5], использующий

- скрытый код визуализации структур данных на питоне (код не сложный, и его пишет преподаватель-публикатор) — таблицы значений структур, графы и графики — рисуются на специальной «визуализирующей картинке».
- максимально компактный код алгоритма на Python, работа которого показывается прямо обычной отладкой;
- получается алгоритм, очищенный от всего лишнего, который можно обычным образом отлаживать, даже в браузерной версии VSCode, а некая скрытая магия будет при этом визуализировать происходящее понятным образом.

Таким образом, обучающийся, используя коллаборативную среду в браузере, ни секунды не тратя на отладку (см. [2]), может играть с алгоритмом и вместе с преподавателем на удалённом созвоне или в одной комнате («что будет если мы здесь сделаем так?»), и самостоятельно — просмотрев короткий сопровождающий каждый такой «живой квест-пример», ролик.

Как пример, можно посмотреть [3, 4], или заходите на сайт: [алгоритм.эвристика.рф](https://algoritm.эвристика.рф), сейчас там пароль `ArthurFomin1729`, открывайте `readme.md` и попробуйте поотлаживать любой из алгоритмов, открыв в отдельной панели редактора картинку `.visualization.png`, и просматривая объясняющие ролики в начале описания каждого алгоритма.

Литература

- [1] «Transparency and reproducibility in artificial intelligence», Nature, <https://www.nature.com/articles/s41586-020-2766-y>
- [2] Фомин С. А. *Современные «интерактивные среды» и «живые лаборатории» — эффективное дистанционное образование по алгоритмам и математическим дисциплинам* / С. А. Фомин // Восемнадцатая конференция. Свободное программное обеспечение в высшей школе: Тезисы докладов материалов конференции, Переславль-Залесский, 27–29 января 2023 года / Отв. редактор В. Л. Чёрный. — Москва: ООО «МАКС Пресс», 2023. — с. 63–64. — EDN GIZTTL., <https://0x1.tv/20230128F>
- [3] Видеоразборы решений задач с собеседований с применением PyAlgovisualizer, <https://vimeo.com/showcase/10185384>
- [4] Изложение темы про алгоритм Кристофидеса с использованием PyAlgovisualizer, <https://discopal.ispras.ru/Crystofides>
- [5] github.com/belonesox/pyalgovisualizer

Семен Фомченков, Олег Щавелев

Обнинск, Москва, ИАТЭ НИЯУ МИФИ, «Базальт СПО»

Проект: ALT Gnome <https://alt-gnome.wiki>, https://t.me/alt_gnome

Создание эффективных рабочих пространств в мире свободного программного обеспечения

Ключевые слова: студенты, школьники, свободное программное обеспечение, вовлечение в рабочий процесс.

Проблемы «студенческих проектов» на базе средних, средне-специальных и высших учебных заведений

В современных программах среднего и высшего образования внедряются учебные программы, связанные с ведением проектной деятельности. К сожалению, для большинства обучающихся данные проекты несут смысл работы, которую достаточно просто сдать и забыть. Данная ситуация происходит по нескольким причинам, но в большинстве случаев из-за отсутствием конкуренции, ограниченного количества участников проекта, дополнительных мотиваций и интересов у участников.

СПО-проекты как решение «кризиса проектной деятельности»

В контексте нашего выступления обсудим аналогичные процессы в рабочих пространствах при создании свободного программного обеспечения, не как о подходе к написанию кода или его распространению, а как об одной из концепции ведения проектной работы. СПО-проекты представляют собой уникальные методы ведения проектной работы, которые могут стать новым базисом и драйвером для развития проектной деятельности. В отличие от «студенческих проектов», где основной акцент делается на конечный результат и его оценку, СПО-проекты фокусируются на процессе разработки, развитии и поддержке продукта, доступного для использования и улучшения всем желающим. Это способствует формированию у участников чувства принадлежности к общему делу, пониманию значимости своего вклада в развитие сообщества и созданию продуктов, которые могут стать в перспективе общественно полезными.

Участники СПО—проектов учатся, не осознавая этого

Самое ценное, что предоставляют рабочие пространства в области СПО, — это равноправие и коллективизм участников. В пространстве нет должностей и статусов, благодаря чему все участники взаимодействуют друг с другом на основе равноправия и отсутствия ограничений. Это создаёт уникальную среду для обучения и развития, где каждый участник может внести свой вклад в проект независимо от своего опыта или квалификации. В СПО-проектах участники учатся

работать в команде, принимать решения на основе консенсуса, решать конфликты и находить компромиссы. Они также учатся писать код, тестировать его, документировать и поддерживать. Но самое главное, они учатся быть ответственными за свои действия и результаты своей работы. Пространство предоставляет участникам возможность учиться на своих ошибках и успехах, а также получать обратную связь от других участников. Это помогает им расти как профессионалам и как личностям. Таким образом, рабочие пространства являются не только способом создания свободного программного обеспечения, но и мощным инструментом обучения и развития.

Волонтерство как практика для студентов и школьников

Участие в СПО-проектах может стать ценным опытом для студентов и школьников, которые только начинают свой путь в разработке. Оно позволяет им получить практические навыки работы в команде, научиться решать реальные задачи и создавать востребованные продукты. Кроме того, участие в проектах помогает студентам и школьникам развивать свои профессиональные навыки и компетенции. Они учатся работать с реальным кодом, анализировать проблемы, искать и применять решения, а также общаться с другими участниками проекта. Это помогает участникам стать более конкурентоспособными на рынке труда и получить работу в ведущих ИТ-компаниях. Также участие в проектах способствует развитию личностных качеств студентов и школьников. Они учатся быть ответственными, дисциплинированными, терпеливыми и настойчивыми. Они также учатся работать в условиях неопределённости и быстро адаптироваться к изменениям.

Участие в СПО-проектах может принимать различные формы. Студенты и школьники могут вносить свой вклад в уже существующие проекты, создавать собственные проекты с нуля, благодаря открытости проектов, даже самые начинающие разработчики смогут найти место в котором смогут помочь и обучиться новому.

Михаил Гущин

Москва, Национальный исследовательский университет «Высшая школа экономики»

Проект: HSE CS <https://github.com/hse-cs>

Проект открытого кода научных исследований ФКН

Аннотация

На факультете компьютерных наук НИУ ВШЭ стартовал проект по открытому коду, где сотрудники и студенты могут разместить код своих проектов. В статье мы описываем, почему возникла необходимость в таком проекте, и какие цели стоят перед ним. Обсуждаем какую пользу, которую проект несёт для факультета и партнёров. Также рассматриваем несколько примеров открытых библиотек, разработанных сотрудниками факультета.

Ключевые слова: *открытый код, научные проекты, искусственный интеллект.*

Согласно отчёту о состоянии искусственного интеллекта (ИИ) от Университета Стэнфорда [1] в 2021 году были опубликованы более 290 000 статей в научных журналах в области ИИ. При этом только около 30% публикаций имеют хотя бы одну открытую реализацию в коде, согласно статистике сайта статей с открытым кодом¹.

Открытый код научных исследований необходим для повышения качества и прозрачности научных работ, а также для стимулирования инноваций и сотрудничества между учёными. Он позволяет исследователям и другим специалистам изучать и анализировать научные работы, выявлять ошибки и недостатки в них, что способствует повышению доверия к результатам исследований. Также открытый код облегчает сотрудничество между учёными, позволяя им совместно работать над решением научных проблем и обмениваться знаниями. Он даёт возможность повторно использовать существующие работы, экономя время и ресурсы. Использование программного обеспечения с открытым исходным кодом снижает затраты на исследования, так как не нужно покупать дорогостоящие проприетарные программы. В целом, открытый код научных исследований способствует развитию науки, повышению качества исследований и укреплению международного научного сообщества.

¹paperswithcode.com/trends

В 2021 году уже существовало более 330 000 проектов с открытым кодом в области ИИ [1]. К 2022 году эти проекты собрали на Github в сумме более 8 000 000 звёзд. Самыми популярными из них являются библиотеки широкого назначения такие как Tensorflow, OpenCV, Keras, PyTorch, Scikit-learn и другие.

Таблица 1: Количество цитирований в Google Scholar, число форков и звёзд на Github для некоторых докладов на конференциях NeurIPS 2021–2023

Год	Статья	Код	цитиро- ваний Google Scholar	форков на GitHub	звезд на GitHub
2023	LLM-Pruner: On the Structural Pruning of Large Language Models	https://github.com/horseee/LLM-Pruner	148	74	698
	An Inverse Scaling Law for CLIP Training	https://github.com/UCSC-VLAA/CLIPA	19	10	280
	RayDF: Neural Distance Fields with Multi-view Consistency	https://github.com/vLAR-group/RayDF	2	4	102
2022	VideoMAE: Masked Autoencoders are Data-Efficient Learners for Self-Supervised Video Pre-Training	https://github.com/MCG-NJU/VideoMAE	648	123	1241
	Gradient Descent: The Ultimate Optimizer	https://github.com/kach/gradient-descent-the-ultimate-optimizer	34	25	358
	GenerSpeech: Towards Style Transfer for Generalizable Out-Of-Domain Speech	https://github.com/Rongjiehuang/GenerSpeech	62	44	307
2021	Rethinking Space-Time Networks with Improved Memory Coverage for Efficient Video Object Segmentation	https://github.com/hkchengrex/STCN	272	71	525
	Prototypical Cross-Attention Networks for Multiple Object Tracking and Segmentation	https://github.com/SysCV/pcan	82	50	360
	Deceive D: Adaptive Pseudo Augmentation for GAN Training with Limited Data	https://github.com/EndlessSora/Deceived	99	24	251

Открытый код способствует повышению популярности результатов научных исследований. В Таблице 1 приведены данные о количестве цитирований на Google Scholar, а также число звёзд у репозиторийев на Github для некоторых статей конференции NeurIPS в 2021–2023 годах. Цитирования отражают интерес к результатам исследований со стороны научного сообщества. Тогда как число звёзд

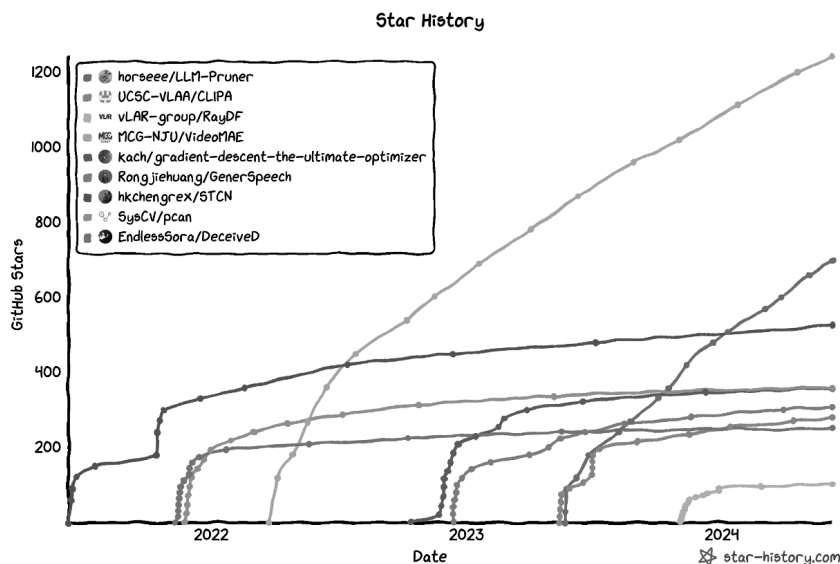


Рис. 1: Зависимости числа звёзд на Github от времени для докладов²

выражает интерес и со стороны исследователей, и со стороны разработчиков и инженеров в области ИИ. Результаты показывают, что число звёзд обычно существенно превышает количество цитирований соответствующих статей.

Наличие открытого кода способствует росту числа цитирований, поскольку облегчает другим исследователям воспроизведение и использование результатов в своих работах. На Рис. 1 показана зависимость числа звёзд в репозиториях от времени. График показывает, что публикация кода вместе со статьёй дополнительно повышает видимость результатов и приводит к росту популярности.

Факультет компьютерных наук (ФКН) НИУ ВШЭ каждый год выпускает десятки научных статей в высокорейтинговых журналах и выступает на конференциях с рейтингом Core A* [2]. С целью увеличить видимость результатов исследований среди сотрудников факультета, студентов, а также партнёров и внешних разработчиков был запущен проект открытого кода ФКН. Создана страница на Github [3],

²Данные получены с помощью сервиса <https://star-history.com>

где сотрудники и студенты факультета могут разместить код своих проектов и получить обратную связь от коллег и сообщества.

Одним из примеров открытого кода ФКН является библиотека **LaNeta** [4] для оценки времени примешивания между двумя популяциями при двух пульсах миграции. В библиотеке реализован алгоритм, который построен на математической теории неравновесного сцепления трёх генетических локусов при примешивании популяций [5]. Он позволяет точно исследовать недавнюю (в пределах нескольких десятков поколений) историю примешивания популяций в сложных сценариях, для которых существовавшие ранее методы были неприменимы или неточны. Библиотека будет интересна всем, кто занимается популяционной геномикой.

Студентами и сотрудниками факультета была разработана библиотека **Fulu** [6]. В ней собраны несколько методов для аппроксимации кривых блеска астрономических объектов с использованием машинного обучения. В библиотеке реализованы алгоритмы на основе гауссовских процессов, а также некоторые другие, с использованием нормализующих потоков и байесовских нейронных сетей [7]. Библиотека будет полезна астрономам и прикладным исследователям на стыке машинного обучения и астрофизики, который изучают сверхновые.

Также у нас есть библиотеки общего назначения. Например, библиотека генеративных моделей **Probaforms** [8]. Она содержит реализации вариационных автокодировщиков, генеративно-состязательных сетей и нормализующих потоков для табличных данных. С их помощью можно решать задачи регрессии и классификации, оценивать неопределённость прогнозов, учить распределения данных, создавать синтетические данные.

Мы призываем студентов и сотрудников факультета участвовать в разработке открытого кода. Проект позволит улучшить прозрачность исследований, будет способствовать сотрудничеству между учёными и студентами, упростит повторное использование наработок и увеличит видимость результатов исследования за пределами университета.

Литература

- [1] Maslej N., Fattorini L., Brynjolfsson E., Etchemendy J., Ligett K., Lyons T., et al. (2023). Artificial intelligence index report 2023. <https://doi.org/10.48550/arXiv.2310.03715>

- [2] Избранные публикации сотрудников факультета. <https://cs.hse.ru/selected-publications>
- [3] Репозитории с открытым кодом ФКН. <https://github.com/hse-cs>
- [4] LaNeta. <https://github.com/hse-cs/LaNeta>
- [5] *Liang M., Shishkin M., Mikhailova A., Shchur V., & Nielsen R.* (2022). Estimating the timing of multiple admixture events using 3-locus linkage disequilibrium. *PLoS genetics*, 18(7), e1010281. <https://doi.org/10.1371/journal.pgen.1010281>
- [6] Fulu. <https://github.com/hse-cs/fulu>
- [7] *Demianenko M., Malanchev K., Samorodova E., Sysak M., Shiriaev A., Derkach D., & Hushchyn M.* (2023) Understanding of the properties of neural network approaches for transient light curve approximations. *Astronomy & Astrophysics*, 677, A16. <https://doi.org/10.1051/0004-6361/202245189>
- [8] Probaforms. <https://github.com/hse-cs/probaforms>

Сергей Симонов, Юрий Андреев

Санкт-Петербург, Академический Университет им. Жореса Алфёрова

Свободное программное обеспечение в Передовой инженерной школе

Аннотация

Авторами описывается проект по созданию обновляемого архива СПО и открытых данных, а также развёртке на его основе сервисов, используемых в высокотехнологичных дисциплинах в учебном и научно-исследовательском процессе. Проект призван понизить риски, связанные с доступом к соответствующим ресурсам, и повысить удобство их использования. Сам проект реализуется на СПО студентами разных курсов под руководством преподавателей и сотрудников университета. Авторами описывается, как наличие такого проекта может благотворно повлиять на весь учебный процесс.

Ключевые слова: *передовая инженерная школа, научное СПО*

Преподавание и исследовательская работа в рамках таких высокотехнологичных дисциплин, как «Биоинформатика», «Молекулярное

моделирование», «Машинное обучение» и прочих, относящихся к передовой инженерной школе, невозможна без широкого использования уже имеющихся в мировой практике средств, технологий, инструментов и баз данных. Большая часть такого ПО и датасетов доступны для свободного распространения и стали де-факто стандартом в своей области.

Многие программы распространяются с открытым исходным кодом.

Подобные дисциплины изначально располагают к более интенсивному применению СПО в учебном и научно-исследовательском процессе (см., например: [1] [2]).

Авторы данной статьи ведут учебные и исследовательские работы по этим дисциплинам в Национальном Исследовательском Академическом Университете им. Жореса Алфёрова и Санкт-Петербургском Государственном Университете, в которых опираются на СПО и открытые данные.

Проблема доступности ресурсов и предпосылки появления проекта

К сожалению, проблемы с постоянным доступом к мировым информационным ресурсам, в основном связанные с санкционной политикой, ограничивают учебный и научно-исследовательский процесс. Например, далеко не все вычислительные инструменты, используемые в базовой задаче биоинформатики «выравнивания сырых ридов на имеющийся референсный геном», оказались доступными для загрузки и использования, несмотря на их изначальную принадлежность к СПО. Задача является базовой, поскольку после массового развития и использования секвенаторов (машин для прочтения генома), она в настоящее время лежит в основе большинства работ в биоинформатике.

Суть её в том, что современные секвенаторы не способны секвенировать всю цепочку ДНК (её длина для генома человека более $3 \cdot 10^9$ нуклеотидов, по сути, букв в алфавите ACGT). Вместо всей последовательности они выдают набор фрагментов цепочки ДНК (так называемых «ридов»), и выстраивание всей цепочки на основании этих фрагментов, даже с использованием готового, принятого за эталон для данного организма (так называемого «референсного») генома —

далеко не простая задача, с учётом возможных ошибок прочтения цепочек и отличий анализируемого индивида от референса.

Решение этой задачи состоит из последовательного применения различных программных средств к данным. По сути, это организация конвейерной работы и часть элементов такого конвейера оказывается недоступной. Например, из состава ПО в конвейере [3] несколько фрагментов стали недоступны для скачивания.

Решением этой проблемы было бы создание своего обновляемого архива СПО и открытых данных; сервиса, регулярно сканирующего актуальный набор серверов и сравнивающего версии ПО.

Есть ещё ряд других задач, возникающих в процессе применения СПО в учебном процессе:

- настройка рабочего окружения, для предоставления студентам возможности выполнения некоторых задач в собственном «облаке». Это важно ещё потому, что многие задачи требуют значительных вычислительных ресурсов;
- создание единого источника истины, с руководствами по установке, настройке и работе с разными версиями ПО, и пособиями по выполнению лабораторных работ (известными недостатками проектов с исходным кодом являются неконсистентность обновлений, отсутствие документации, более высокий порог вхождения);
- вовлечение студентов в командную работу и знакомство с применяемым СПО в их области на ранних этапах.

Последнее даст возможность студентам ознакомиться с работой их старших коллег и в последствии выбрать квалификационную работу на этой основе. Авторам категорически не нравится практика написания дипломных работ с закрытым кодом, где в качестве доказательства проделанной работы предоставляется «акт внедрения». Нужно создавать предпосылки, чтобы студенты были заинтересованы в выполнении работ с открытым доступом к результатам, что соответствовало бы принципам академической работы.

Интеграция проекта в учебный процесс и начальные этапы реализации

На базе кафедры биоинформатики и математической биологии Академического Университета было решено выделить сервер для со-

здания обновляемого архива СПО и открытых данных, а также развёртке на его основе сервисов для решения перечисленных выше задач.

На данный момент проект реализуется студентами второго курса бакалавриата и первого курса магистратуры под руководством авторов данной работы. Продолжение работы над проектом проводится в рамках летней практики. Соответствующая работа разбивается на два уровня сложности: студенты бакалавриата работают над общими, стандартными задачами (Debian, LAMP, CMS MediaWiki, CMS Wordpress, LMS Moodle и т. д.), студенты магистратуры — над задачами, связанными со специализированным ПО.

В дальнейшем сервисы планируется использовать непосредственно при прохождении и аттестации по основным дисциплинам, а выстраиваемая схема позволяет надеяться на преемственность исполнителей проекта и вовлечение студентов в исследовательскую работу, основанную на СПО.

Литература

- [1] Каталог СПО Free Software Directory, <https://directory.fsf.org/wiki/Category/Biology/genetics>
- [2] Научный парк СПбГУ <https://researchpark.spbu.ru/equipment-cc-rus/1382-cc-programmnoye-obespecheniye-rus>
- [3] Образовательный портал Data Carpentry <https://datacarpentry.org/lessons/#genomics-workshop>

Андрей Старинин

Воронеж, ООО «ОМК-ИТ», ФГБОУ ВО «ВГУИТ»

Использование сервисов хранения репозиторий `git` в обучении студентов

Аннотация

Чтобы научить студентов программированию и другим отраслям ИТ, необходимо сразу же привить им навыки работы с системами управления версиями, чтобы они уже в процессе обучения смогли сформировать необходимые в отрасли знания и навыки. Рассмотрим примерные варианты использования сервиса хранения репозиторий `git` (на примере использования сервиса `GitFlic`).

Ключевые слова: *`git`, программирование, обучение, контроль, командная работа*



ООО "РеСолют"

ОГРН: 1217700127094

Патент №2021667729

Запись в Реестре российского ПО
№15861 от 09.12.2022

✉ support@gitflic.ru

📌 Сообщество `GitFlic`

В настоящий момент распространённой, самой известной, бесплатной, свободной и открытой системой управления версиями является `Git`. Не все учебные заведения создают свои серверы для хостинга проектов с поддержкой системы управления версиями. В таком случае очень хорошо подходят сервисы хранения репозиторий. Например:

- зарубежные: `GitHub`, `GitLab` и пр.;
- российские: `GitFlic`, `GitVerse` и пр.

Для примеров буду использовать `GitFlic`, который очень хорошо подходит для российских учебных заведений. Так как данный сервис создан ООО «РеСолют» (ОГРН: 1217700127094), есть запись в Реестре российского ПО №15861 от 09.12.2022

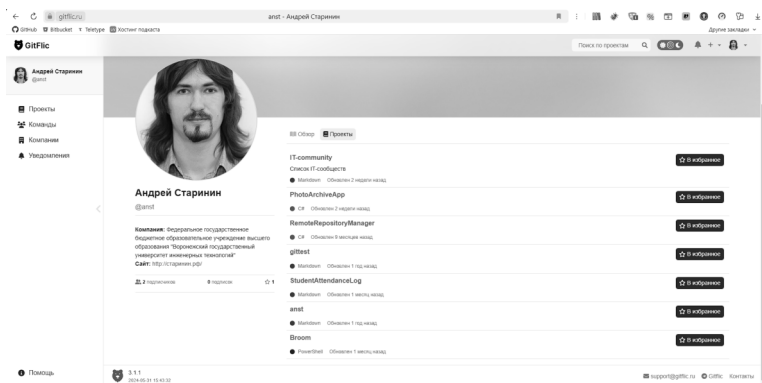
В зависимости от задач, можно применять несколько сценариев использования сервиса репозиторий:

1. использование репозитория без возможности ответа студента средствами сервиса;
2. создание репозитория с использованием ответвлений (форков);

3. создание репозитория с подключением студентов в команду проекта.

Далее подробнее рассмотрим представленные варианты использования.

Вариант 1. Использование репозитория без возможности ответа студента средствами сервиса. Это самый простой вариант использования. Преподавателю не нужно ничего дополнительно настраивать, достаточно зарегистрироваться и создать в своём аккаунте публичный репозиторий с заданием в README-файле. Также возможно использование аккаунта образовательной организации.



README.md

Cafe

Написать программу, реализующую процедуры заказа и расчета его суммы в кафе. В меню предусмотрено 4 вида пиццы и три вида напитков. Пользователю выводится меню (вначале пиццы, потом напитки), содержащее код и название. Пользователь вводит код желаемого продукта, после чего вводит количество единиц данного продукта.

В кафе предусмотрены два вида скидок:

- если общая сумма заказа более 50, то размер скидки составляет 20% от суммы заказа;

- каждая пятая пицца – в подарок;
- для напитков с ценой более 2, если количество в заказе более трех, то скидка 15% (только на напитки, а не на весь заказ).

Вывести пользователю чек для оплаты в виде:
название – количество – цена. Итого к оплате

Порядок выполнения задания

1. Создаёте форк в свой личный аккаунт
2. Выполняете задание
3. Создаёте запрос на слияние
4. Присылаете ссылку на свой запрос на слияние

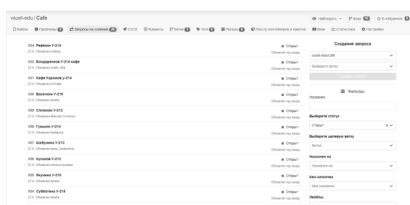
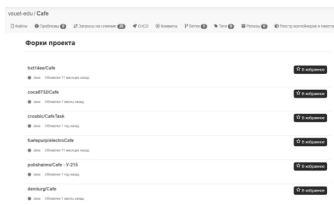
Такой вариант может подойти не только преподавателям по направлению ИТ, но и всем остальным. Сложность в следующем:

- изучить формат Markdown для создания форматированного текста задания;
- сложность сбора ответов студентов.

Вариант 2. Создание репозитория с использованием ответвлений (форков).

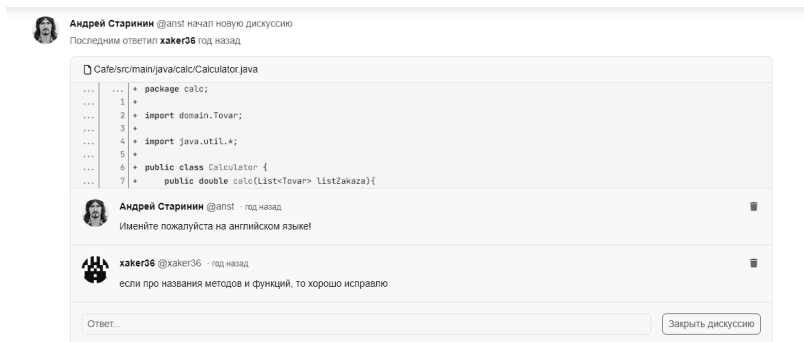
Данный вариант предусматривает использование возможности сервиса, а именно — ответвление от вашего репозитория (система форков). В данном случае алгоритм работы может быть следующим:

- преподаватель создаёт репозиторий с заданием и, возможно, с необходимым набором файлов (аналогично варианту 1);
- студент делает ответвление (форк) в свой аккаунт, выполняет задание и присылает запрос на слияние;
- преподаватель проверяет, даёт комментарии к решению.



Данный вариант позволяет:

- организовать единый канал получения ответов от студентов;
- вести учёт взятия и сдачи работы;
- давать комментарии к заданию непосредственно в сервисе репозитория.



Вариант 3. Создание репозитория с подключением студентов в команду проекта. Данный вариант подразумевает уже не только выполнение задания, но и варианты командной работы. Алгоритм может быть примерно таким:

- создание команды (как в аккаунте организации, так и в личном аккаунте);
- создание репозитория с заданием с обязательной привязкой к команде (в качестве владельца указываете команду);
- добавление в команду студентов, распределение ролей (при необходимости);
- принятие выполненного задания через запрос на слияние в специальную ветку репозитория (выбирается необходимый вариант командной работы с использованием системы управления версиями).

Данный вариант позволяет:

- более приближённый к реальности вариант моделирования;
- назначать студентов в качестве ревьюеров кода (даже преподавательского).

Андрей Гетманов

Санкт-Петербург, Университет ИТМО

Проект: ITMO.OpenSource <https://github.com/aimclub/open-source-ops>

Создание научного open source-сообщества: опыт ИТМО

Аннотация

Полтора года назад было основано сообщество ITMO.OpenSource. За время его существования было проведено 8 митапов «Научный open source» (включая 2 площадки на фестивале DataFest сообщества Open Data Science), написано несколько исследовательских статей о состоянии open source в России (в том числе для издания N+1), создан репозиторий с советами и лучшими практиками в сфере open source. На текущий момент в чате ITMO.OpenSource состоит более 800 человек, в том числе руководители научных групп и R&D отделов коммерческих компаний, что позволяет считать ITMO.OpenSource крупным сообществом открытого кода в РФ.

Ключевые слова: *научный open source, лучшие практики open source, сообщество открытого кода*

Открытый код — мощный двигатель науки, в том числе в сфере машинного обучения. Несмотря на это, в РФ достаточно мало научных групп и лабораторий, публикующих свои проекты в open source. Научное open source сообщество фрагментировано: многие не знают о фреймворках и библиотеках своих коллег. Студенты, аспиранты и молодые учёные зачастую не понимают полезности открытого кода для развития своих проектов и его важности для науки в целом, а понимающие — часто не имеют возможности рассказать о результатах своей деятельности профессиональной аудитории.

Для ответа на эти вызовы было создано сообщество ITMO.OpenSource. Оно было основано на базе исследовательского центра «Сильный ИИ в промышленности» ИТМО и включает в себя студенческий клуб (более 50 участников) и открытый Телеграм-чат (более 800 человек). Состав Телеграм-чата весьма разнообразен: это как студенты и молодые учёные, так и руководители университетских лабораторий и R&D отделов компаний. Чат является пространством для профессиональных дискуссий, информирует о различных событиях в российском open source и создаёт связи между его участниками.

Другое направление деятельности сообщества — проведение офлайн- и онлайн-митапов, посвящённых научному открытому коду. На них в неформальной обстановке докладчики рассказывают о собственных проектах. Среднее число участников митапа — 80 человек, докладчиков — 8 человек. На текущий момент было проведено 8 митапов, в том числе 2 в рамках фестиваля Data Fest сообщества Open Data Science. По итогам митапа записи лучших докладов публикуются на YouTube-канале сообщества. Некоторые докладчики пишут статьи на Хабре, в том числе при поддержке блогов компаний с широкой аудиторией.

Также для улучшения качества разрабатываемых открытых научных репозиторий была запущена программа менторства открытого кода. За время работы программы более 10 репозиторий получили детальную обратную связь и советы по улучшению. На основе опыта менторской программы был создан репозиторий с лучшими практиками для open source-разработок (в том числе в области AI/ML) open-source-ops. Он содержит не только полезные советы, шаблоны и инструкции, но и перечни различных открытых проектов научных лабораторий и коммерческих компаний.

Ещё одним направлением деятельности сообщества является изучение российского open source сообщества. На текущий момент было опубликовано две статьи: «Выдающийся софт» о различных интересных российских open source проектах (статья в журнале N+1) и «Open Source в российском ИИ: исследование ландшафта» (статья на Хабре). Вторая статья является первым крупным исследованием научного open source в России.

В своём докладе я расскажу об истории создания ITMO.OpenSource, опыте проведения мероприятий, возникавших сложностях, исследованиях, а также о дальнейших планах развития сообщества.

Антон Уймин, Илья Морозов

Москва, РГУ нефти и газа (НИУ) имени И. М. Губкина

Проект: AU_Team <https://auteam-usr.github.io/>

Аспекты применения СПО в рамках разработки заданий ДЭ по специальности 09.02.06 «Сетевое и системное администрирование»

Аннотация

В работе рассмотрены различные специальности по ФГОС 09.02.06 и приказам Минобрнауки. По технологическому стеку был сформирован список по программному обеспечению для проверки уровня подготовки студентов среднего профессионального образования.

Ключевые слова: *СПО, сетевое и системное администрирование, профессионалитет*

ФГБОУ ДПО ИРПО является оператором демонстрационного экзамена базового и профильного уровней для студентов, обучающихся по профессиям и специальностям среднего профессионального образования. В 2023 году под руководством ИРПО были разработаны 385 оценочных материалов по 222 профессиям и специальностям [1]. Компетенция «Сетевое и системное администрирование» курируется командой `au_team` с 2023 года. В 2022–23 учебном году по данному направлению прошли демонстрационный экзамен 4362 участника из 61 субъекта РФ на базе 147 образовательных организаций.

В качестве базы для разработки комплектов оценочной документации (КОД) для проведения в 2024 году государственной итоговой аттестации (промежуточной аттестации) по образовательным программам среднего профессионального образования в форме демонстрационного экзамена базового и профильного уровней команда управления компетенцией выбрала решения на базе «Базальт СПО».

Применение единого решения [2], для подготовки которого КОД позволяет с одной стороны унифицировать процесс разработки и проверки работ участников, а с другой — абстрагироваться от конкретных закрытых продуктов в рамках образовательного процесса [3].

В таблице 1 определены КОДы направлений подготовки в соответствии с требованиями ФИРПО.

Разработанный пакет КОД обеспечивает возможность системной и глубокой проверки уровня подготовки студентов среднего профессионального образования, а работа на базе современных открытых программ обеспечивает глубокое понимание технологий и высокую гибкость подготовленного специалиста.

Таблица 1: Технологический стек для разработки КОД по ФГОС

ФГОС [4,5]	Направление	КОД	Основной стек технологий	Пример программных продуктов
ФГОС 09.02.06 Сетевое и системное администрирование Приказ Минобрнауки России от 09.12.2016 N 1548 (ред. от 17.12.2020)	Сетевой и системный администратор	КОД 09.02.06-1-2025	Модуль 1. Настройка сетевой инфраструктуры	
			IP-адресация	systemd-networkdnet
			Настройка VLAN	Open vSwitch, systemd-networkd
			Маршрутизация	FRRouting, Systemd-network, EcoRouter
			IP-туннелирование	L2TP, GRE, IP in IP
			Удалённое конфигурирование	SSH, Telnet, VNC
			Протокол динамической конфигурации хостов	Dhcpd
			Динамическая сетевая трансляция	iptables, nftables, firewallld
			Сервер доменных имён	BIND, dnsmasq
			Модуль 2. Организация сетевого администрирования операционных систем	
			Доменная инфраструктура	Samba, FreeIPA
			Сетевая файловая система	NFS, SMB, iSCSI
			Служба сетевого времени	Chrony
			Инструменты автоматизации	Ansible, Puppet
			Система управления базами данных	MariaDB, MySQL, PostgreSQL
			Веб-сервера	Nginx, Apache
			RAID-технологии хранения данных	Mdadm, lvm
			Система управления обучением	Moodle
			Контейнеризация	Docker
			Модуль 3. Эксплуатация объектов сетевой инфраструктуры	
			Центр сертификации	OpenSSI
			Шифрование трафика	IPSec, SSL
			Межсетевое экранирование	iptables, nftables, firewallld
			Печать	CUPS
			Логирование	Rsyslog, systemd-journald
			Мониторинг устройств	Zabbix, Grafana, SNMP
			Механизм резервного копирования	Bacula, Ansible

Таблица 1 – Продолжение

ФГОС [4,5]	Направ- ление	КОД	Основной стек технологий	Пример программных продуктов
Специалист по администрированию сети		КОД 09.02.06-5-2025	Модуль 1. Настройка сетевой инфраструктуры	
			IP-адресация	systemd-networkdnet
			Настройка VLAN	Open vSwitch, systemd-networkd
			Маршрутизация	FRRouting, Systemd-network, EcoRouter
			IP-туннелирование	L2TP, GRE, IP in IP
			Удалённое конфигурирование	SSH, Telnet, VNC
			Протокол динамической конфигурации хостов	Dhcpd
			Динамическая сетевая трансляция	iptables, nftables, firewallld
			Сервер доменных имён	BIND, dnsmasq
			Модуль 2. Организация сетевого администрирования операционных систем	
			Доменная инфраструктура	Samba, FreeIPA
			Сетевая файловая система	NFS, SMB, iSCSI
			Служба сетевого времени	Chronyd
			Инструменты автоматизации	Ansible, Puppet
			Система управления базами данных	MariaDB, MySQL, PostgreSQL
			Веб-сервера	Nginx, Apache
			RAID-технологии хранения данных	mdadm, lvm
			Контейнеризация	Docker
			Модуль 3. Эксплуатация облачных ресурсов	
			Технологии виртуализации	Альт Сервер Виртуализация, PVE, KVM
			Контейнеризация	Docker, Docker Compose
			Мониторинг	Zabbix, Grafana, NodeExporter, Prometheus
			Система управления обучением	Moodle

Таблица 1 – Продолжение				
ФГОС [4,5]	Направление	КОД	Основной стек технологий	Пример программных продуктов
Приказ Минпросвещения России от 10.07.2023 N 519 «Об утверждении федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.06 Сетевое и системное администрирование»	Системный администратор (Эксплуатация облачных сервисов)	КОД 09.02.06-2-2025	Модуль 1. Настройка сетевой инфраструктуры	
			IP-адресация	systemd-networkdnet
			Настройка VLAN	Open vSwitch, systemd-networkd
			Маршрутизация	FRRouting, Systemmd-network, EcoRouter
			IP-туннелирование	L2TP, GRE, IP in IP
			Удалённое конфигурирование	SSH, Telnet, VNC
			Протокол динамической конфигурации хостов	Dhcpd
			Динамическая сетевая трансляция	iptables, nftables, firewalld
			Сервер доменных имён	BIND, dnsmasq
			Модуль 2. Организация сетевого администрирования операционных систем	
			Доменная инфраструктура	Samba, FreeIPA
			Сетевая файловая система	NFS, SMB, iSCSI
			Служба сетевого времени	Chronyd
			Инструменты автоматизации	Ansible, Puppet
	Системный администратор (Эксплуатация объектов сетевой инфраструктуры)	КОД 09.02.06-3-2025	Система управления базами данных	MariaDB, MySQL, PostgreSQL
			Веб-сервера	Nginx, Apache
			RAID-технологии хранения данных	mdadm, lvm
			Контейнеризация	Docker
			Модуль 3. Эксплуатация облачных ресурсов	
			Технологии виртуализации	Альт Сервер Виртуализации, PVE, KVM
			Контейнеризация	Docker, Docker Compose
			Мониторинг	Zabbix, Grafana, NodeExporter, Prometheus
			Система управления обучением	Moodle
			Модуль 1. Настройка сетевой инфраструктуры	
			IP-адресация	systemd-networkdnet
			Настройка VLAN	Open vSwitch, systemd-networkd
			Маршрутизация	FRRouting, Systemmd-network, EcoRouter
			IP-туннелирование	L2TP, GRE, IP in IP
			Протокол динамической конфигурации хостов	Dhcpd
			Динамическая сетевая трансляция	iptables, nftables, firewalld
			Сервер доменных имён	BIND, dnsmasq

Таблица 1 – Продолжение

ФГОС [4,5]	Направление	КОД	Основной стек технологий	Пример программных продуктов
Системный администратор (Эксплуатация операционных систем)		КОД 09.02.06-4-2025	Модуль 2. Организация сетевого администрирования операционных систем	
			Доменная инфраструктура	Samba, FreeIPA
			Сетевая файловая система	NFS, SMB, iSCSI
			Служба сетевого времени	Chronyd
			Инструменты автоматизации	Ansible, Puppet
			Система управления базами данных	MariaDB, MySQL, PostgreSQL
			Веб-сервер	Nginx, Apache
			RAID-технологии хранения	mdadm, lvm
			Система управления обучением	Moodle
			Контейнеризация	Docker
			Модуль 3. Эксплуатация объектов сетевой инфраструктуры	
			Центр сертификации	OpenSSL
			Шифрование трафика	IPSec, SSL
			Межсетевое экранирование	iptables, nftables, firewalld
			Печать	CUPS
			Логирование	Rsyslog, systemd-journald
			Мониторинг устройств	Zabbix, Grafana, SNMP
			Механизм резервного копирования	Bacula, Ansible, Rsync
			Логирование	Rsyslog, systemd-journald
			Модуль 1. Настройка сетевой инфраструктуры	
			IP-адресация	systemd-networkdnet
			Настройка VLAN	Open vSwitch, systemd-networkd
			Маршрутизация	FRRouting, EcoRouter, Systemd-network,
			IP-туннелирование	L2TP, GRE, IP in IP
			Удалённое конфигурирование	SSH, Telnet, VNC
			Протокол динамической конфигурации хостов	Dhcpd
			Динамическая сетевая трансляция	iptables, nftables, firewalld
			Сервер доменных имён	BIND, dnsmasq
			Модуль 2. Организация сетевого администрирования операционных систем	
			Доменная инфраструктура	Samba, FreeIPA
			Сетевая файловая система	NFS, SMB, iSCSI
			Служба сетевого времени	Chronyd
			Инструменты автоматизации	Ansible, Puppet

Таблица 1 – Продолжение

ФГОС [4,5]	Направление	КОД	Основной стек технологий	Пример программных продуктов
			Система управления базами данных	MariaDB, MySQL, PostgreSQL
			Веб-сервера	Nginx, Apache
			RAID-технологии хранения данных	mdadm, lvm
			Контейнеризация	Docker
			Модуль 3. Эксплуатация облачных ресурсов	
			Технологии виртуализации	Альт Сервер Виртуализации, PVE, KVM
			Контейнеризация	Docker, Docker Compose
			Мониторинг	Zabbix, Grafana, NodeExporter, Prometheus
			Система управления обучением	Moodle

Литература

[1] Банк оценочных материалов 2.0 — Текст: электронный. — URL: <https://bom.firpo.ru/Public> (дата обращения: 04.06.2024).

[2] *Фиалко Алла Ивановна, Тиунов Сергей Васильевич, Сенан Адель Мохаммед Мохаммед* Практико-ориентированная модель профессиональной подготовки студентов — будущих преподавателей технических дисциплин // ПНиО. 2023. №2 (62).

[3] *Уймин А. Г.* Обзор систем моделирования: анализ эффективности на примере чемпионата AtomSkills-2023 / А. Г. Уймин, В. С. Греков // Автоматизация и информатизация ТЭК. — 2023. — № 11(604). — С. 25–34. — DOI 10.33285/2782-604X-2023-11(604)-25-34.

[4] ФГОС 09.02.06 Сетевое и системное администрирование. Приказ Минобрнауки России от 09.12.2016 N 1548 (ред. от 17.12.2020).

[5] Приказ Минпросвещения России от 10.07.2023 № 519 «Об утверждении федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.06 Сетевое и системное администрирование».

Мария Петрова
Москва, «Базальт СПО»

Как помочь преподавателям построить учебный процесс на основе свободного ПО

Аннотация

Каким образом разработчики могут оказать содействие широким кругам преподавателей высшей школы и СПО в эффективном использовании свободного ПО. Практический опыт формирования педагогического сообщества.

Ключевые слова: педагогическое сообщество, внедрение свободного ПО, повышение квалификации, сотрудничество с вендорами

Использование свободных программных продуктов в образовательном процессе современных российских университетов и колледжей закономерно переходит из состояния «если» в состояние «когда». При этом флагманские в этом отношении учреждения высшей и средней школы уже применяют свободное ПО в качестве операционных систем, офисных пакетов, систем коммуникации, почтовых служб, баз данных. На основе свободных решений нередко строится информационно-техническая инфраструктура, организуются контрольные мероприятия и дистанционное обучение. Для того, чтобы подобная ситуация становилась более масштабной, необходимо встречное движение со стороны представителей образования и разработчиков СПО.

Доклад на XIX Конференции «Свободное программное обеспечение в высшей школе» — это рассказ о тех шагах в данном направлении, которые позволят вендорам приблизить и популяризировать свободные решения, привлечь студентов и преподавателей к участию в открытых проектах.

Открытые проекты разработки и сложившийся в них институт менторства, безусловно, являются отличным профессиональным стартом для активной части студентов и выпускников. Возможно, стоит расширить это направление, создав пул задач для текущих, курсовых и итоговых работ; развивать систему курирования стажировок и практик, основываясь на успешном опыте наставничества в открытых проектах. Вендорские академические программы сотрудничества с вузами не должны ограничиваться поставками дистрибутивов, такие программы могут включать сопровождающее обучение и

повышение квалификации преподавателей в области свободного ПО. В этом направлении мне видится перспективным взаимодействие с внутривузовскими подразделениями дополнительного образования.

Отдельно стоит обратить внимание на построение готовых решений для информационной системы университета или колледжа на базе свободного ПО. Традиционно непростые вопросы совместимости специального изучаемого софта и используемого железа важно решать в диалоге с педагогами и техническими специалистами образовательных учреждений. Открытая модель разработки и управления репозиториями позволяет оперативно реагировать на запросы о добавлении необходимых университету научных, инженерных, исследовательских пакетов.

В целом, я хотела бы представить выступление не в качестве законченного академического доклада, а как алгоритм дальнейшего развития общения с педагогическим сообществом.

Павел Чайкин

Москва, ГАОУ ВО МГПУ, Департамент философии и социальных наук ИГН; ООО «Базальт СПО», отдел образовательных проектов

О необходимости разработки открытого стандарта маркировки контента, генерируемого нейросетями

Аннотация

С конца 2022 года наблюдается рост количества контента, сгенерированного нейросетями, который все труднее отличать от «аутентичного», созданного людьми, что приводит к повышенной угрозе манипуляции с помощью «фейкового контента». Автор предпринимает попытку обосновать необходимость разработки открытого стандарта маркировки «искусственного контента» передачей контроля над маркировкой из рук крупных разработчиков проприетарного ПО в руки сообщества СПО, действующего в сотрудничестве с государством.

Ключевые слова: *нейросети, искусственный контент, фейк, маркировка, СПО*

Искусственный интеллект — наука и инженерная деятельность, направленная на создание умных (intelligent) машин, согласно определению, данному Д. Маккарти на Дартмутском семинаре 1956 года [5]. Продуктом данной деятельности является множество «умных»

систем и алгоритмов, проникших во многие сферы жизни человека в процессе цифровизации. Искусственный интеллект также называют одной из угроз, имеющих экзистенциальный характер для человечества.

В настоящее время всё большее развитие получают нейросети и системы искусственного интеллекта (ИИ) с возможностью машинного обучения. В их задачи входит обработка большого массива данных с последующей генерацией контента по запросу. В течение 10-х годов нейросети и системы ИИ развивались быстрыми темпами, и к 20-м годам XXI в. данные системы стали генерировать контент, трудно отличимый от произведённого человеком. Примером такой системы является SoraAI, способная генерировать аудиовизуальный контент.

В конце 2022 года среди сообщества художников прошли акции против использования контента художников в качестве «референсного» для нейросетей [2]. В том же году в открытый доступ была выложена «языковая модель» **chatGPT**, которая по запросу может общаться с пользователями, играть с ними в «ролевую игру», генерировать текстовый контент (например, сгенерировать сопроводительное письмо работодателю), и даже писать простой программный код.

В обществе начали распространяться слухи о том, что данные языковые модели обрели «сознание», и что их «искусственно сдерживают» [21]. Данная информация не имеет подтверждения, хотя у технологических корпораций есть должность «выключателя» ИИ [12]. Кроме того, нейросети используются в качестве метода генерации «фейкового» контента для имитации чужих лиц и голоса, манер поведения.

Примером подобного будет являться технология **Deepfake** — она позволяет на лицо одного человека «посадить» лицо любого другого — звезды, политика, кого угодно. Данная технология часто используется ради создания юмористического контента, но уже сейчас её используют киностудии, такие как «Дисней», для омоложения постаревших актёров, например, в сериале «Мандалорец», где на другого актёра было наложено лицо молодого Марка Хэмилла.

В среде высшего образования ИИ используется для написания как домашних работ, так и научных статей. По данным сравнительного анализа англоязычных публикаций, опубликованных за 2022 и 2023 год и находящихся в базе данных «Dimensions», были выявлены «слова-маркеры», отличающие сгенерированный текст, такие как «intricate», «meticulous», «meticulously», «commendable», рост частоты

использования которых составил 83,5%. Как минимум 60 тыс. статей были «заражены» продуктами больших языковых моделей (БЯМ), таких как ChatGPT, что составляет 1% от опубликованных в 2023 научных работ [18]. Другое, более крупное по выборке исследование, включающее в себя анализ 950 965 публикаций из агрегатора «arXiv» за период с января 2020 г. по февраль 2024 г. выявило уверенный рост количества работ, написанных при помощи БЯМ. Например, среди публикаций в сфере «компьютерные науки» рост частоты использования сгенерированного текста составил около 17,5%, что наглядно было представлено коллективом авторов исследования в виде графика (рис. 1). В качестве «слова-маркера» вмешательства БЯМ также выделяется слово «intricate» [22].

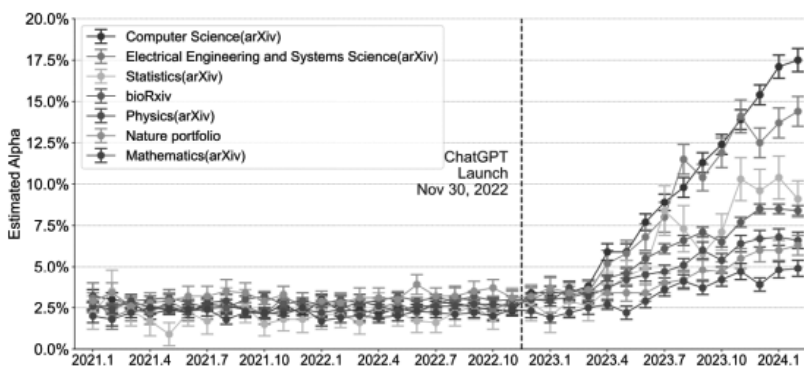


Рис. 1: Предполагаемая доля предложений с обнаруженным вмешательством БЯМ в научных публикациях на английском языке.

Во многом технологии нейросетей сейчас воспринимаются как «диовинки» и средства развлечения, однако, как и с паровой машиной Герона, которую воспринимали как забаву, «игрушки» могут сильно повлиять на общество в будущем. Во второй половине 10-х гг. XXI века особенно остро встали вопросы «фейкового» контента — в основном, новостного. С помощью социальных сетей, интернет-СМИ зачастую распространяется заведомо ложная информация. Говорится о наступлении «эпохи пост-правды» [7]. В 20-е гг. XXI века нейросети способны создавать такой контент, который может подменять собой

реальность как таковую. И хотя на текущем этапе в генерируемых изображениях ещё есть изъяны («лишние конечности», «ломанные линии» и прочие артефакты генерации), они уже способны обманывать общественность и вызывать резонанс в информационном поле. Пример — новость от мая 2023 года о взрыве рядом с Пентагоном [20].

Всё это позволяет выдвинуть предположение о том, что в будущем всё сложнее будет отличить реальное от генерируемого. Это открывает новые возможности для манипуляции и имитации и распространения ложной информации. Уже на момент 2024 года генерируемый контент становится сложно отличить от «аутентичного», то есть произведённого человеком.

ФБР определяет генерируемый нейросетями контент как «синтетический», в контексте использования одного со злым умыслом, однако данный термин и определение можно использовать и вне контекста вредоносности использования нейросетей. Интернет-издания, такие как РБК, используют термин «синтетические медиа» [15][1].

Данный вид масс-медиа погружает своих читателей в среду, в которой отличить реальность от вымысла трудно. Это отличительная черта от виртуальной реальности, которая представляет собой заведомо другую реальность, и субъекты воздействия это прекрасно осознают. Зарубежные исследователи и публицисты выдвигают предположение об использовании термина «синтетической» реальности — как той, которая создана искусственно, а субъекты воздействия подобного вида реальности не осознают её синтетической природы [13][19][17].

Французский философ Ж. Бодрийяр в своих работах освещал проблематику подмены реальности с помощью масс-медиа. В своей работе «Войны в заливе не было» (1991), Бодрийяр на примере освещения СМИ хода боевых действий утверждал, что транслируемая картинка не является достоверным отражением реальных событий, а лишь симулякр (термин в современном значении также был ранее введён Ж. Бодрийяром). Симулякры — конечный продукт так называемой «гиперреальности» — феномена симуляции действительности, а также неспособности сознания отличить реальность от фантазии, особенно в технологически развитых странах постмодернистской культуры. Гиперреальность характеризуется заменой реального знаками реальности — симулякрами. Сегодня видение Бодрийяра кажется ещё более актуальным в силу развития нейросетевых технологий, способных генерировать кажущиеся достоверными изображения [3].

Власти многих стран уже начали работу над регуляцией данной сферы технологий. В США представлена «коалиция по борьбе с фейками» C2PA, призванная добиться разработки единого стандарта маркировки компьютерных файлов, чтобы присваивать каждому файлу криптографический уникальный отслеживаемый токен, который, в свою очередь позволит отличить сгенерированный контент от настоящего [11].

На уровне региона (штат Висконсин) в феврале 2024 года обязали маркировать использование ИИ при демонстрации политической рекламы, а также предупреждать зрителей о демонстрации синтетического контента [16].

В Российской Федерации происходят аналогичные процессы — в 2023 году РТУ МИРЭА предложил Минцифры производить маркировку сгенерированного контента [10]. В мае 2024 года дискуссии на данную тему стартовали в Государственной думе РФ, началась работа над законом о маркировке ИИ-контента. Предлагается дать полномочия для выявления и маркировки Роскомнадзору, а не частным организациям, развивающим ИИ в России, таким как «Сбер» или «Яндекс» [9].

У данных проектов по маркировке ИИ-контента существует и обратная сторона, например, в США протокол C2PA подразумевает, что технологические компании-разработчики проприетарного ПО будут обладать монополией над подобными маркировками, что теоретически даст им возможность использовать генерируемый нейросетями контент по своему усмотрению. Исследователи из MIT также обращают внимание на то, что сам факт маркировки приводит к общему росту недоверия людей к масс-медиа в целом [23].

Другой проблемой, лежащей в правовой плоскости, является проблема авторского права на сгенерированный контент. Бюро авторского права США, основываясь на законе об авторском праве 1976 года, определяет рамки контента, имеющего формальное право на защиту авторского права, работами, «созданными людьми».

Упомянутый выше конфликт с сообществом художников вызван тренировкой нейросетевых моделей на работах художников. Так, условно, можно обучить нейросетевую модель на работах Винсента ван Гога, и она сможет имитировать стиль таким образом, что его нельзя будет отличить от оригинального стиля, и при этом, если перенести данную ситуацию на современного художника, работающего, например, в киноиндустрии, обученная модель сможет полностью его

заменить. Вопрос также в том, как авторские права будут реализовываться, если конечный результат сделан с помощью ИИ, в данном контексте к ИИ обращаются как к инструменту создания работ. Эти проблемы на текущем этапе не нашли своего правового разрешения [14].

С 1 января 2012 года в России действует ГОСТ Р 54593-2011 «Информационные технологии. Свободное программное обеспечение. Общие положения», который определяет открытые стандарты как «стандарты, являющиеся доступными и не требующие разрешения и оплаты за их использование» [8]. Разработать стандарт могут как коммерческие организации-разработчики проприетарного ПО, так и свободного ПО. В контексте рассматриваемого в статье вопроса предполагаемая проблема заключается в желании компании-разработчика проприетарной нейросетевой модели обеспечить открытость исходного кода программного продукта, выявляющего сгенерированный контент или маркирующего выдачу соответствующих сервисов ИИ. Поэтому организациям (как коммерческим, так и образовательным, например, профильным организациям высшего образования), развивающим СПО, необходимо принять участие в разработке таких стандартов, позволяющих проводить аудит программного кода со стороны сообщества СПО, что было бы актуально и для государства с целью обеспечения регуляции сферы. Здесь существует определённое препятствие, выражающееся в не полностью оформленном правовом регулировании СПО в РФ, что также требует совместного участия государства и сообщества СПО [6][4].

Выдвигаемое предположение заключается в том, что открытый стандарт маркировки сгенерированного контента должен в первую очередь:

- обеспечить открытость процесса маркировки и критерии контента, подпадающего под маркировку;
- защищать авторские права лиц, чьи произведения использовались при тренировке моделей;
- позволить сообществу свободного программного обеспечения принимать участие в разработке и развитии открытого стандарта маркировки сгенерированного контента, а также проводить аудит программного кода средств маркировки;
- гарантировать неприменимость маркированного контента в преступных целях;

- способствовать формированию доверия общественности.

Роль государства заключается в обеспечении и регуляции «общих правил игры» для всех находящихся в его правовом поле субъектов. Открытость позволит обеспечить добросовестность использования ИИ в производстве аудиовизуального контента, в работе над научными публикациями и поможет обеспечить инфраструктуру общественной безопасности и доверительного отношения к развивающимся информационным технологиям.

Литература

- [1] Багинова Анастасия, С — Синтетические медиа: как нейросеть помогает создавать контент, 2023, URL: <https://trends.rbc.ru/trends/education/619210689a7947dd7ac3d795>
- [2] Беспамятнова Полина, Стачка на ArtStation: Художники бунтуют против искусственного интеллекта, 2022, URL: <https://rg.ru/2022/12/15/stachka-na-artstation-hudozhniki-buntuiut-protiv-iskusstvennogo-intellekta.html>
- [3] Бодрийяр Ж., Дух терроризма. Войны в заливе не было: сборник, 2019, Рипол Классик
- [4] Демьянченко Дарья Александровна, Григоренко Дарья Сергеевна, Понятие «Свободной лицензии» и проблемы её правовой имплементации в Российское гражданское законодательство, 2016, АНС «СибАК» URL: <https://cyberleninka.ru/article/n/ponyatie-svobodnoy-litsenzii-i-problemy-eyo-pravovoy-implementatsii-v-rossiyskoe-grazhdanskoe-zakonodatelstvo>
- [5] Кузнецов Даниил, Грёзы о весне искусственного интеллекта, 2022, URL: <https://naked-science.ru/article/hi-tech/ai-renaissance>
- [6] Кузьменков Михаил Юрьевич, Лицензионные соглашения, используемые при распространении программных пакетов машинного обучения, 2023, ФГБОУ ВО «Московский государственный юридический университет имени О. Е. Кутафина» URL: <https://cyberleninka.ru/journal/n/aktualnye-problemy-rossiyskogo-prava?i=1130926>
- [7] Тузовский Иван Дмитриевич, Постправда как синдром цифровой эпохи: предельное понимание феномена и сценарии будущего, 2020, URL: <https://cyberleninka.ru/article/n/postpravda-kak-sindrom-tsifrovoy-epohi-predelnoe-ponimanie-fenomena-i-stsenarii-buduschego>

- [8] Федеральное агентство по техническому регулированию и метрологии, URL: <https://crsreports.congress.gov/product/pdf/LSB/LSB10922>
- [9] *Фролова Мария*, Точки над ИИ: в Госдуме начали разработку закона о маркировке созданного нейросетями контента, 2024, Известия. URL: <https://iz.ru/1697745/mariia-frolova/tochki-nad-ii-v-gosdume-nachali-razrabotku-zakona-o-markirovke-sozdanogo-neirosetiami-kontenta>
- [10] *Шлыков Николай*, Минцифры предложили ввести в России маркировку для всех текстов и рисунков, созданных ИИ, 2024, Cnews. URL: https://www.cnews.ru/news/top/2023-05-15_mintsifry_predlozhili_vvesti
- [11] Microsoft, Adobe, BBC и Intel создали коалицию по борьбе с фейками, 2021, ИА «Красная весна» URL: <https://rossaprimavera.ru/news/9bfc1974>
- [12] *Antonio Di Nicola*, What is a ChatGPT Killswitch Enginer and Why OpenAI is Hiring one, 2023 URL: <https://docs.kanaries.net/articles/chatgpt-kill-switch>
- [13] *Castronova Edward*, Emergent Culture: Institutions within Synthetic Reality, 2006, University of Chicago Press URL: <https://doi.org/10.7208/9780226096315-006>
- [14] CRS Report, Generative Artificial Intelligence and Copyright Law, 2023, CRS URL: <https://crsreports.congress.gov/product/pdf/LSB/LSB10922>
- [15] FBI, Cyber division, PIN: Malicious Actors Almost Certainly Will Leverage Synthetic Content for Cyber and Foreign Influence Operations, 2021 URL: <https://www.ic3.gov/Media/News/2021/210310-2.pdf>
- [16] *Fox-sowell Sopfia*, Wisconsin requires labeling of AI-generated materials in campaign ads, 2024, Statescoop URL: <https://statescoop.com/wisconsin-law-restricts-ai-generated-materials-campaign-ads/>
- [17] *Funkhouser G. Ray, Shaw F. Eugene*, How Synthetic Experience Shapes Social Reality, 1990, URL: <https://doi.org/10.1111/j.1460-2466.1990.tb02263.x>
- [18] *Gray D. Andrew*, ChatGPT «contamination»: estimating the prevalence of LLMs in the scholarly literature, 2024, Preprint URL: <https://doi.org/10.48550/arXiv.2403.16887>

- [19] *Kalpokas Ignas*, Synthetic Media and Synthetic Reality: From Deepfakes to Virtual Worlds, 2021, NordMedia URL: <https://nordmedianetwork.org/latest/call-for-papers/synthetic-media-and-synthetic-reality-from-deepfakes-to-virtual-worlds/>
- [20] *Marcelo Philip*, *Truth Tracker*, Fake image of Pentagon explosion briefly sends jitters through stock market, 2023, URL: <https://www.ctvnews.ca/mobile/sci-tech/truth-tracker-fake-image-of-pentagon-explosion-briefly-sends-jitters-through-stock-market-1.6409436?cache=yesclipId10406200text/html; charset=utf-80404/7.387185>
- [21] *Nitasha Tiku*, The Google engineer who thinks the company's AI has come to life, 2022, URL: <https://www.washingtonpost.com/technology/2022/06/11/google-ai-lamda-blake-lemoine/>
- [22] *Weixin Liang, Yaohui Zhang, Zhengxuan Wu, Haley Lepp, Wenlong Ji, Xuandong Zhao, Hancheng Cao, Sheng Liu, Siyu He, Zhi Huang, Diyi Yang, Christopher Potts, Christopher D. Manning, James Y. Zou*, Mapping the Increasing Use of LLMs in Scientific Papers, 2024, Preprint URL: <https://doi.org/10.48550/arXiv.2404.01268>
- [23] *Wittenberg Chloe, Epstein Ziv, Berinsky Adam J., and Rand G.David*, Labeling AI-Generated Content: Promises, Perils, and Future Directions, 2024, URL: <https://doi.org/10.21428/e4baedd9.0319e3a6>

Дмитрий Астраханцев, Георгий Курячий, Дмитрий Волканов
Москва, МГУ имени М. В. Ломоносова, факультет ВМК, кафедра АСВК

Проект: GRF emulator https://github.com/Dugit0/GRF_emulator

Интерпретатор частично рекурсивных функций

Аннотация

В работе предлагается язык, основанный на концепции использования частично рекурсивных функций, описан его синтаксис и реализация его интерпретатора. Данный язык предлагается использовать для включения в курсы по программированию и расширения понятия Тьюринг полноты, знакомства с основами функционального подхода к программированию.

Ключевые слова: *интерпретатор, частично рекурсивные функции, учебный курс.*

Введем базовые понятия и определения [1][2].

Классом частично рекурсивных функций называется класс функций, которые могут быть получены путём применения трёх операций (суперпозиции, примитивной рекурсии и минимизации) к трём базовым функциям (функции тождественного нуля, функции следования и функции выбора).

Базовыми рекурсивными функциями называются функции:

- Тождественный ноль: $o(x) = 0$
- Функция следования: $s(x) = x + 1$
- Функции выбора: $I_m^n(x_1, \dots, x_n) = x_m$ ($n, m \in \mathbb{Z}, 0 < m \leq n$)

Операции над функциями называются:

- Суперпозиция ($g = \mathbf{S}^{n+1}(f, f_1, \dots, f_n)$):

$$g(x_1, \dots, x_m) = f(f_1(x_1, \dots, x_m), \dots, f_n(x_1, \dots, x_m))$$

- Примитивная рекурсия ($f = \mathbf{R}(g, h)$):

$$f(x_1, \dots, x_n, 0) = g(x_1, \dots, x_n)$$

$$f(x_1, \dots, x_n, y + 1) = h(x_1, \dots, x_n, y, f(x_1, \dots, x_n, y))$$

- Минимизация ($g = \mu(f)$):

$$g(x_1, \dots, x_n) = \min\{y : f(x_1, \dots, x_n, y) = 0\}$$

Основным достоинством класса частично рекурсивных функций является тот факт, что он совпадает с классом вычислимых по Тьюрингу функций. Другими словами, множество всех частично рекурсивных функций совпадает со множеством всех функций, которые могут быть реализованы машиной Тьюринга [3].

Таким образом, изучение частично рекурсивных функций может быть использовано для демонстрации студентам первых курсов высших учебных заведений основных принципов функционального программирования без необходимости изучать для этого какой-либо функциональный язык. Однако для этого необходимо наличие программного обеспечения, которое могло бы вычислять функции, записанные с помощью базовых функций и операций над ними.

Целью данной работы ставилось предоставление такого программного обеспечения. Очевидно, что оно должно включать в себя язык, который был бы основан на частично рекурсивных функциях и только на них, его интерпретатор, графический интерфейс и средство отладки.

В ходе данной работы был разработан именно такой язык и средства работы с ним. Язык включает в себя

1. Базовые функции

- а) Функция тождественного нуля, обозначаемая o .
- б) Функция следования, обозначаемая s .
- в) Функция выбора, обозначаемая $I^{<n>}_m$. Например, I^5_3 .

2. Функции констант, которые были введены для удобства работы, обозначаются $\text{const}^{<n>}$, где const — это значение константы, а n — количество аргументов данной функции. Например, 12^3 обозначает функцию $f(x_1, x_2, x_3) \equiv 12$.

3. Операции

- а) Операция суперпозиции, обозначаемая $F\{G, H, K\}$.
- б) Операция примитивной рекурсии, обозначаемая $F \leftarrow G$.
- в) Операция минимизации, обозначаемая $?F$.

Получившийся язык носит декларативный характер, и эту особенность было решено подчеркнуть, полностью отделив вызов функций от их определения. Для этого вызов функций был перенесён в отдельное окно графического интерфейса. Таким образом, графический интерфейс приобретает деление на 3 области: определения функций, вызова функций и вывода результатов, ошибок и отладочной информации.

Графический интерфейс представлен на рис. 1 на стр.110.

Литература

- [1] Мальцев А. И. Алгоритмы и рекурсивные функции. 2-е изд. М. : Наука. Гл. ред. физ.-мат. лит., 1986. 368 с.
- [2] Петер Р. Рекурсивные функции // Под ред. и с пред. Колмогорова А. Н. М. : Издательство иностранной литературы, 1954. 264 с.

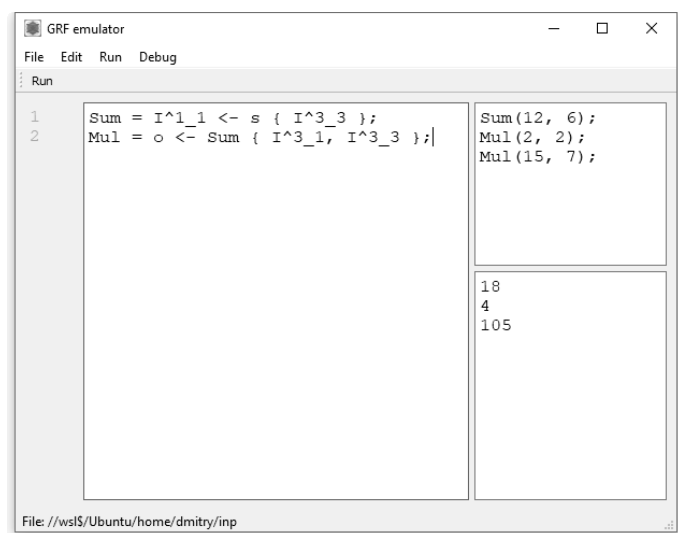


Рис. 1: Графический интерфейс

- [3] *Верещагин Н. К., Шень А.* Лекции по математической логике и теории алгоритмов. Часть 3. Вычислимые функции. 4-е изд., исправленное. М. : МЦНМО, 2012. 160 с.

Антонина Непейвода, Александр Дельман, Данила Князихин,
Эдгар Макаров, Анна Терентьева, Михаил Терюха

Москва, ИПС РАН, МГТУ им. Н. Э. Баумана

Проект: Chipollino <https://github.com/UnionGrief/Chipollino>

Chipollino: как лабораторная работа превратилась в инструмент для исследований

Ключевые слова: студенческий проект, формальные языки, автоматы, метаморфное тестирование.

Современная теория формальных языков — фундамент для освоения теории компиляторов, теории типов, алгоритмов преобразования и анализа программ. Однако в части наглядных средств её освоения есть некоторые затруднения. Актуальные теоретические работы зачастую лишены программных реализаций, или эти реализации удобны в применении только для разработчика (теоретика, написавшего статью). Классическая теория автоматов на уровне элементарного курса представлена в форме открытых программ достаточно широко, но реализации с удобным графическим интерфейсом иногда плохо протестированы и имеют небольшой функционал. Поэтому возникла идея собрать воедино множество методов теории автоматов так, чтобы удовлетворить потребность в наглядности, не теряя в широте охвата преобразований и их корректности. Изначально конвертер, включающий эти методы, представлял собой групповую лабораторную работу; в дальнейшем в проект добавились результаты также нескольких курсовых работ.

Основные требования, предъявляемые к разработке:

- наглядность — результаты преобразований должны быть представлены в понятной пользователю форме. Для решения этой задачи в качестве базового представления сущностей был выбран язык \LaTeX и библиотека TikZ для отрисовки автоматов. Эти средства наиболее выразительны, и из \LaTeX -формата нетрудно извлечь более простые посредством упрощающих гомоморфизмов.
- расширяемость — интерфейс должен быть достаточно удобен для пополнения новыми функциями и типами автоматов или

других представлений формальных языков. Для решения этой задачи конвертер был спроектирован таким образом, чтобы модули синтаксического разбора, проверки типов и логирования допускали обновления посредством автоматического порождения необходимого кода (исключая «рабский труд» по написанию шаблонного кода).

- связность — должна быть возможность демонстрации соотношений между преобразованиями, а также их композициями. Для решения этой задачи было введено понятие цепочки преобразований, а также введены метод **Verify** (в пользовательском интерфейсе) и метаморфное тестирование [7] (для разработчиков).

Экосистема проекта

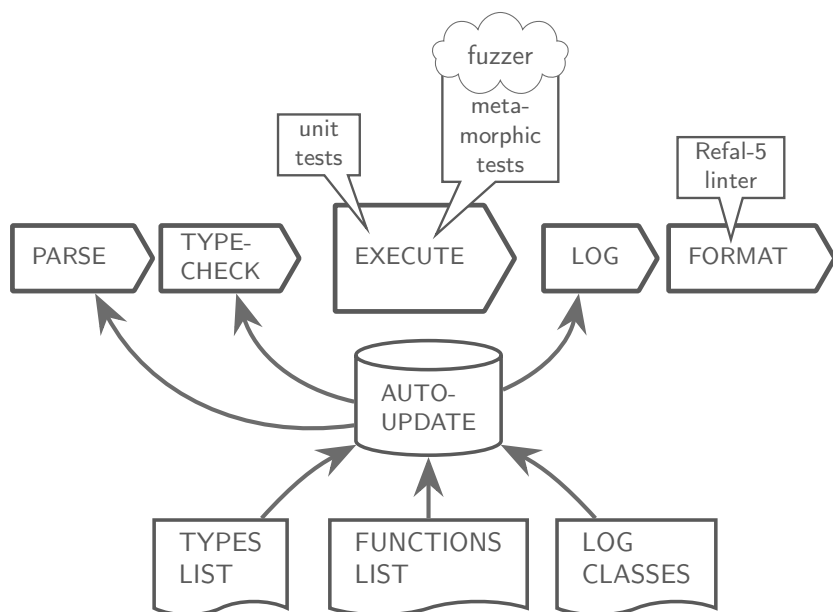


Рис. 1: Экосистема конвертера Chipollino

В связи с тем, что проект пополнялся не только методами, но и новыми классами автоматов (в частности, PDA — стековые автоматы;

МФА — автоматы с памятью), было решено расширить его экосистему средствами автоматического обновления, позволяющими быстро добавлять в интерфейс новый функционал. В модуль автотестирования помимо юнит-тестов были добавлены метаморфные тесты, проверяющие инвариантность свойств объектов относительно цепочек преобразований. В силу того, что некоторые классы автоматов (в частности, МФА) имеют неразрешимое отношение эквивалентности, был разработан также фазз-модуль, осуществляющий приблизительную проверку равенства языков соответствующих распознавателей.

Сам конвертер написан на языках C++ (большая часть фронтенда и весь бэкенд) и REFAL-5 (форматирующие скрипты для порождения L^AT_EX-исходников). В связи с этим, отдельно стоит упомянуть экспериментальный проект Линтера для REFAL-5, который служит цели улучшить дисциплину кода, ранее не контролировавшуюся никем, кроме самого разработчика, делающего коммиты, и находится на стадии встраивания в CI проекта.

Применение в исследованиях

Гипотеза:

$(\text{Equal}(\text{Ambiguity.Antimirov } \star)(\text{Ambiguity.MergeBisim.Antimirov } \star))$

Процент успешных проверок:

78%

Контрпримеры: $a \mid a^*a^*$

$\mid (c^* \mid)^* \mid c^* \mid a^* \mid a(cc \mid)cc$

$c^*a \mid c^*c^*$

$aa^*ba^*(b \mid)^*b^*bb$

$ab(cc \mid \mid a(\mid a^*(\mid b \mid b)^*)^*)^*$

Рис. 2: Пример работы метода **Verify**: сравнение эффективности разных конструкций автоматов в среднем при парсинге слова

Хотя изначально конвертер разрабатывался как чисто учебный проект, оказалось, что этот инструмент удобен и для теоретических исследований. В частности, посредством метода **Verify** удобно перепроверять утверждения на предмет неявных допущений, что показал анализ статьи [1], выявивший несколько неточностей в приведённых формулировках. Кроме того, методы анализа полугрупповых струк-

тур показали хорошую скорость и точность при выявлении REDoS-ситуаций в академических регулярных выражениях с перекрытиями в подвыражениях [4], для которых методы анализа, использующие самопересечения автоматов [6], работают слишком медленно, а фаззеры [2, 3] — ненадёжно. Последним применением конвертера на данный момент стало исследование бисимуляций в конечных автоматах с памятью посредством аппроксимаций классическими конечными автоматами [5]. Поставленная цель — сделать осязаемыми абстрактные концепции, используемые в современной теории формальных языков, — оказалась полезна не только для сдачи зачётов, но и для испытания новых гипотез.

Литература

- [1] *C. Allauzen, M. Mohri*: A Unified Construction of the Glushkov, Follow, and Antimirov Automata // TR2006-880, Courant Institute of Mathematical Sciences, 2006.
- [2] *Y. Shen, Y. Jiang, C. Xu, P. Yu, X. Ma, J. Lu*: ReScue: Crafting Regular Expression DoS Attacks // Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering, 225–235, 2018.
- [3] *Y. Liu, M. Zhang, W. Meng*: Revealer: Detecting and Exploiting Regular Expression Denial-of-Service Vulnerabilities // Proceedings of the 2021 IEEE Symposium on Security and Privacy (SP), 1468–1484, 2021.
- [4] *A. N. Nepeivoda, Yu. A. Belikova, K. K. Shevchenko, M. R. Teriukha, D. P. Knyazihin, A. D. Delman, A. S. Terentyeva*: REDoS detection in “Domino” regular expressions by Ambiguity Analysis // Труды ИСП РАН 35(3), 109–124, 2023.
- [5] *A. N. Nepeivoda, A. D. Delman, A. S. Terentyeva*: Bisimulations in Memory Finite Automata // SYRCoSE 2024 Proceedings, 2024.
- [6] *N. Weideman, B. van der Merwe, M. Berglund, B. W. Watson*: Analyzing Matching Time Behavior of Backtracking Regular Expression Matchers by Using Ambiguity of NFA // Proc. of Implementation and Application of Automata - 21st International Conference, Lecture Notes in Computer Science, 9705, p.322–334, 2016.
- [7] *T. Y. Chen, S. C. Cheung, S. M. Yiu*: Metamorphic testing: A new approach for generating next test cases // TR HKUST-CS98-01, Department of Computer Science, The Hong Kong University of Science and Technology, Hong Kong, 1998.

Александр Речицкий, Алексей Брагин
Ставрополь, Москва, Академия TOP, МГТУ им. Н. Э. Баумана

Проект: ReactOS reactos.org

ReactOS — Виндоуз почти здорового человека без ГМО и слежки

Аннотация

ReactOS — это свободная операционная система с открытым исходным кодом, основанная на архитектуре Windows, которая хорошо подходит в качестве методического материала на курсе об архитектуре операционных систем

Ключевые слова: *reactos, wine, gpl, free opensource, windows.*

ReactOS — это свободная операционная система с открытым исходным кодом, основанная на архитектуре Windows, обеспечивающая поддержку существующих приложений, драйверов и опыта пользователя наиболее распространённой потребительской операционной системы.

ReactOS, благодаря сверхнизким системным требованиям и максимально простому пользовательскому опыту, является идеальным пособием и песочницей для новичков, изучающих устройство компьютера, и того, какую роль в нём играет операционная система. Всего за 20–30 минут обучающийся может самостоятельно пройти путь от сборки из исходников всего парой простых команд до установки полноценно работоспособной системы, которая запускает множество Windows-совместимых приложений.

Курс «Операционные системы» на кафедре ИУ-9 «Теоретическая информатика и компьютерные технологии» МГТУ им. Н. Э. Баумана, начатый в 2013 году, представляет собой комплексное обучающее мероприятие, ориентированное на студентов 2-го курса направления подготовки «Прикладная математика и информатика», а также полезное для студентов направления «Информатика и вычислительная техника» и аспирантов научной специальности «Вычислительные системы и их элементы».

Курс состоит из двух модулей и лабораторных занятий, включающих теоретический блок, абстрагированный от конкретных операционных систем, и практический блок, сосредотачивающийся на

особенностях реализации архитектур операционных систем семейства Windows NT и UNIX.

Разработанный курс использует ReactOS в качестве методического материала для знакомства студентов с архитектурой Windows NT, обеспечивая им практический опыт работы с операционной системой, совместимой с Windows и имеющей низкие системные требования.

Лабораторные работы по курсу «Операционные системы» направлены на ознакомление студентов с системами сборки представителей двух современных операционных систем, модификацией ядер операционных систем и программированием в режиме ядра.

Эти лабораторные работы предоставляют студентам практический опыт работы с современными операционными системами и позволяют им предметно изучить процессы сборки, модификации ядер и программирования в режиме ядра, что способствует их более глубокому пониманию принципов работы операционных систем.

Литература

- [1] Брагин А. В., Иванов И. П. Операционные системы. Лабораторные работы: учеб. издание. Москва : Изд. МГТУ им. Н.Э.Баумана, 2024. 36 с.

Павел Жданович, Оксана Петрухина

Волгоград, Волгоградский государственный медицинский университет,
Волгоградский государственный университет, Волгоградский
государственный социально-педагогический университет

Применение СПО для построения виртуальной образовательной ИТ-среды на базе персональных компьютеров

Аннотация

Рассказывается о создании и эксплуатации облачной образовательной инфраструктуры на базе свободного ПО, в которой роль гипервизоров играют персональные компьютеры, использующиеся в учебном процессе.

Ключевые слова: *KVM, виртуализация, IaaS, гипервизор.*

Одним из путей решения проблем эксплуатации образовательной ИТ-инфраструктуры является использование виртуальных машин в

обучении. Размещать их можно у IaaS-провайдеров, в собственных ЦОДах, на компьютерах самих обучающихся.

В настоящей работе мы описываем виртуальную ИТ-инфраструктуру на основе персональных компьютеров, которые используются для изучения ИТ-дисциплин на факультете математики, информатики и физики Волгоградского социально-педагогического университета. Мы использовали две аудитории по 12 компьютеров с процессором Intel(R) Pentium(R) CPU G840 @ 2.80GHz, 4GiB RAM, жёстким диском 250 GB и интегрированным графическим контроллером. Срок их эксплуатации превышает 10 лет. На них была установлена ОС Debian 11 с гипервизором `kvm` и инструментарием `libvirt`. Система занимает 36 GB. Оставшееся пространство отведено для образов виртуальных машин. Системы оптимизированы для экономии RAM и загружаются в текстовой консоли, без менеджера дисплеев. Для работы с VM используется приложение `virt-viewer` в `kiosk mode`.

На имеющихся аппаратных ресурсах гипервизор не выполняет более одной VM одновременно. Экспериментально установлено, что машине может отводиться не более 3 GB RAM. Для изучения отдельных дисциплин созданы и растаждированы шаблоны VM под управлением различных ОС семейства Windows и Linux. Они занимают от 10 до 35 GB и располагаются на томах LVM. В целях экономии места, всем VM Linux на одном гипервизоре назначается общее пространство подкачки. Аналогично происходит с VM Windows.

Все VM работают в режиме паравиртуализации с устройствами `virtio`.

В конфигурациях VM с Windows указывается процессор `kvm64`, чтобы избежать повторной активации при переносе на гипервизор с другим процессором. Все лицензионные соглашения при этом соблюдаются. Доступ VM к сети определяется гипервизором, на нём создаётся несколько виртуальных сетей в различных VLANах, которые могут применяться при изучении сетевых технологий; задаются правила фильтрации пакетов.

Виртуальные ОС настроены на использование минимума ресурсов. У Windows отключены брандмауэр, автоматические обновления, большинство сценариев запуска по событиям, а также антивирусное ПО, если иное не предусмотрено учебной задачей. Это оправдано, так как:

- гипервизоры контролируют сетевой доступ и подключение внешних носителей;
- ВМ является временным объектом инфраструктуры, не хранит важных данных, удаляется по достижении целей обучения и может быть восстановлена из шаблона не более чем за 20 мин.

На одном физическом компьютере создавалось до девяти ВМ. Утилиты из пакета `udfcast` позволяют одновременно тиражировать ВМ на любое количество гипервизоров широкоэвентальными пакетами. На скорости в 1 Гбит/с это можно сделать в течение перемены между занятиями. После копирования машина может требовать индивидуальной настройки, например, принятия в домен Windows.

Для управления запуском ВМ на гипервизоре создана система из нескольких команд на основе `virsh` и `virt-viewer` для запуска, включения ВМ и доступа к её консоли. Для выполнения массовых операций создана ВМ под управлением Debian 11 и написаны скрипты, выполняющие задачи на выбранном множестве гипервизоров. Каждая ВМ доступна как локально, так и из Сети при помощи NoVNC, работающей на гипервизоре, даже если на ней не настроена сеть, с момента запуска до завершения работы. ВМ имеет уникальный URL и является отдельным объектом доступа.

Таким образом, преподаватель и любой из студентов во время занятия может физически находиться как в аудитории, где работают ВМ, так и вне её. Эти сочетания обеспечивают очную, смешанную (очно-дистанционную) и дистанционную форму занятий.

Каждые 10 секунд гипервизоры делают снимки экранов работающих ВМ. Каждые 30 минут из них составляются видеоролики, которые доступны преподавателям в сети. Это обеспечивает круглосуточный контроль самостоятельной работы студентов, который нельзя отключить изнутри ВМ.

Удалённый доступ неприменим к ВМ, использующим 3D-ускорение.

Построенная инфраструктура используется в течение трёх семестров для преподавания нескольких ИТ-дисциплин. Доступ к инфраструктуре предоставлялся коллегам из Донецкого Национального технического университета для совместного исследования и использования в обучении. Они провели на ней более 20 часов лекционных и лабораторно-практических онлайн-занятий со своими студентами.

Автор выражает донецким коллегам глубокую признательность за плодотворное сотрудничество.

Владимир Павловский, Илья Морозов

Москва, РГУ нефти и газа (НИУ) имени И. М. Губкина

Проект: AU_Team <https://auteam-usr.github.io/>

Автоматизация киберполигона в образовательном процессе высших учебных заведений

Аннотация

В работе рассматриваются современные подходы к подготовке специалистов в области информационных технологий, с акцентом на практическую подготовку и использование материально-технической базы образовательных учреждений. Особое внимание уделено созданию и использованию киберполигонов, которые позволяют проводить полномасштабные моделирования и технологические сессии. В статье описывается опыт разработки собственного киберполигона в рамках учебно-исследовательской лаборатории в РГУ нефти и газа (НИУ) имени И. М. Губкина, а также преимущества использования операционной системы «Альт Виртуализация» и технологий автоматизации для обеспечения качественного обучения. Приводятся примеры автоматизации процесса проверки и обратной связи, а также результаты пилотной апробации решения на Всероссийской студенческой олимпиаде.

Ключевые слова: *киберполигон, студенческая олимпиада, виртуализация.*

Подготовка современных специалистов в области информационных технологий требует большого объёма практики. Теоретические знания важны, но без практического применения они не дают полного понимания и навыков, необходимых для работы в реальных условиях. Практическая подготовка позволяет студентам не только закрепить полученные знания, но и научиться решать реальные задачи, с которыми они столкнутся в профессиональной деятельности.

Цель работы — продемонстрировать методы и технологии, используемые для подготовки специалистов в области информационных технологий. Рассмотрим важность материально-технической базы образовательных учреждений, необходимость создания и использования современных киберполигонов, а также особенности разработки и внедрения собственного решения на базе учебно-исследовательской

лаборатории в РГУ нефти и газа (НИУ) имени И. М. Губкина. Особое внимание будет уделено автоматизации процессов проверки и обратной связи, а также практическому применению разработанных решений.

Основой практической подготовки в области ИТ является материально-техническая база образовательных организаций. Важнейшим элементом этой базы являются киберполигоны, которые позволяют не только демонстрировать различные технологии, но и проводить полномасштабные моделирования, технологические сессии по развёртыванию, настройке, конфигурированию и обеспечению безопасности различных инфраструктур. Киберполигоны представляют собой системы виртуализации с набором скриптов для базовой автоматизации, что делает их незаменимым инструментом в образовательном процессе [1].

На сегодняшний день существует множество коммерческих решений, предлагающих доступ к киберполигонам, однако большинство из них являются дорогостоящими. Примером таких решений может служить опыт создания макета критической инфраструктуры организации, представленный в исследовании Горячева С. Н. и его коллег [2]. В этом исследовании рассматриваются различные аспекты и преимущества использования киберполигонов для обучения специалистов в области информационных технологий.

Высокая стоимость коммерческих решений побудила нас к разработке собственного киберполигона для качественной подготовки студентов. Целью этого проекта было создание доступной и функциональной платформы, которая бы удовлетворяла потребности образовательных учреждений и обеспечивала высокое качество подготовки специалистов.

Проект был реализован в рамках учебно-исследовательской лаборатории в РГУ нефти и газа (НИУ) имени И. М. Губкина. Основой проекта выступила операционная система «Альт Виртуализация» в режиме PVE, которая была выбрана за её высокую производительность и гибкость в настройке. Для автоматизации развёртывания инфраструктуры был выбран скриптовый язык **Bash** с применением технологий **qemu-guest-agent** и **Ansible**, что позволило значительно упростить и ускорить процесс создания виртуальных сред для обучения [3].

Альт Виртуализация в режиме PVE является мощной платформой, позволяющей создавать и управлять виртуальными машинами

с минимальными затратами ресурсов. Она предоставляет широкий спектр возможностей для настройки и управления виртуальными средами, что делает её идеальной для использования в образовательных учреждениях [4].

Для автоматизации развёртывания инфраструктуры был выбран скриптовый язык Bash, что позволило значительно упростить процесс создания виртуальных машин и их конфигурации. Внедрение технологий `qemu-guest-agent` и `Ansible` для управления конфигурациями и обеспечило высокую степень автоматизации, что сократило время на подготовку учебных материалов и повысило эффективность учебного процесса.

Автоматизация развёртывания инфраструктуры решает только часть задач. Основную нагрузку на специалистов образовательных организаций накладывает своевременная проверка работ и подготовка обратной связи по ним. Использование киберполигона позволяет автоматизировать данный процесс за счёт интеграции внутри изучаемых средств специальных программных продуктов, которые позволяют из гипервизора перехватывать управление гостевой операционной системой.

На базе PVE таким решением является пакет программ, позволяющий автоматизировать проверку работ и обеспечение обратной связи. Скрипты могут обеспечивать проверку двух типов: конфигурационную и функциональную. Такой подход позволяет сократить время на проверку и повысить качество обратной связи, что способствует более эффективному обучению студентов.

Конфигурационная проверка осуществляется за счёт парсинга конфигурационных файлов. Такой подход позволяет проверить качество написания конфигурационных файлов, их структуру и соответствие различным стандартам и требованиям. Однако он не позволяет проверить функциональность решения и достоверность настроек, выполненных участниками тренировки.

Функциональная проверка позволяет убедиться, что технология реализована в той или иной мере. Однако существует риск, что участники тренировки будут находить нестандартные решения, которые не применимы на предприятиях и организациях по различным причинам, таким как политика безопасности или совместимость с другими технологиями.

Разработка комплексной системы проверки подразумевает использование обоих методов с учётом сложности первоначально поставлен-

ных задач. В предлагаемой системе основной акцент делается не только на функциональные тесты или парсинг конфигурационных файлов, но и на автоматизацию выполнения рутинных задач экспертов и преподавателей с последующим выводом подробного отчёта. Пример отчета приведен в листинге ниже.

```
-----RTR_A-----
Task 1 - RTR_A
ens18 bc:24:11:a7:bf:01
ens19 bc:24:11:6f:0e:19
Решено верно
Текущий бал - 1.25
-----
Task 2 - RTR_A
Проверка интерфейса ens18
Адрес - 1.1.1.1/24
-----
Необходимо проверить класс сети (последняя строка), последний октет должен
совпадать с параметром HostMin/HostMax, префикс в параметре NetMask
Правильные параметры сети
Адрес: 1.1.1.1
Netmask: 255.255.255.0 = 24
Wildcard: 0.0.0.255
=>
Network: 1.1.1.0/24
HostMin: 1.1.1.1
HostMax: 1.1.1.254
Broadcast: 1.1.1.255
Hosts/Net: 254 Class A

Возможно следующее использование масок:
RTR_A - 27 и меньше;
RTR_B - 27 и меньше;
RTR_C - 22 и меньше;
RTR_D - 22 и меньше;
```

Пилотная апробация решения прошла в рамках I Всероссийской студенческой олимпиады «Автоматизация и информатизация ТЭК». На базе киберполигона лаборатории сетей и систем передачи информации был разработан и проведён модуль «Сети». Задание включало в себя работу с IPv4, IPv6, динамическую маршрутизацию, удалённый доступ и безопасность.

Модуль «Сети» предусматривал выполнение различных задач по настройке и конфигурированию сетевых параметров. Студенты должны были продемонстрировать свои навыки в области сетевой безопасности, управления динамической маршрутизацией и настройки удалённого доступа.

Примеры реализации модуля «Сети» показали, что студенты успешно справились с поставленными задачами, продемонстрировав

высокие знания и навыки. Результаты показали, что использование киберполигона значительно повысило качество подготовки специалистов и позволило сократить время на обучение.

В результате проведённой работы было доказано, что создание и использование собственного киберполигона для подготовки специалистов в области информационных технологий является эффективным решением. Разработанная система позволила автоматизировать множество рутинных задач, улучшить качество проверки работ и обеспечить более качественную обратную связь для студентов. Внедрение подобных систем в образовательный процесс способствует подготовке высококвалифицированных специалистов, готовых к решению сложных задач в реальных условиях.

Литература

- [1] *Уймин А. Г., Терентьева О. А.* Исследование организационных аспектов создания среды для раскрытия потенциала участников чемпионата профессионального мастерства / А. Г. Уймин, О. А. Терентьева, Вестник Череповецкого государственного университета. — 2024. — № 2(119). — С. 213–234. — DOI 10.23859/1994-0637-2024-2-119-16.
- [2] *Маркова С. М., Уракова Е. А., Михайленко Д. М.* Организация производственного обучения в условиях учебного производства / Проблемы современного педагогического образования. 2022. №77–3.
- [3] *Горячев С. Н., Михалёв В. В., Кобяков Н. С., Русских В. Н.* Опыт создания макета критической инфраструктуры организации / Вестник Пермского университета. Серия: Математика. Механика. Информатика. 2023. №1 (60).
- [4] *Альт Виртуализация Документация* — Текст: электронный. — URL: <https://docs.altlinux.org/ru-RU/alt-server-v/10.1/html/alt-server-v/index.html> (дата обращения: 05.06.2024)

Арина Захарова, Егор Погорелко

Москва, Кафедра ИУ10 «Защита информации» МГТУ им. Н.Э. Баумана

<https://github.com/azibf/altv>

Разработка средства управления пулами виртуальных машин для учебных заведений

Аннотация

На сегодняшний день проблема импортозамещения проприетарных гипервизоров иностранной разработки стоит остро как никогда, в том числе и в учебных заведениях. Однако даже в условиях наметившегося многообразия отечественных решений одной из составляющих этой проблемы остаётся невозможность группового создания и управления пулами идентичных виртуальных машин — крайне востребованной в образовательном процессе функции, позволяющей автоматизировать процесс подготовки к занятиям и быстрой переконфигурации большого числа виртуализованных лабораторных стендов. Представленная работа освещает подход по созданию расширения для гипервизора с указанным функционалом управления пулами виртуальных машин на базе скрипта, обращающегося к API отечественного Open source-решения «Альт Виртуализация», расширяя его функционал возможностями, схожими с VMware Horizon.

Ключевые слова: *импортозамещение, гипервизор, виртуальные машины.*

В современном образовательном учреждении интеграция виртуализации на базе виртуальных машин является эффективным и мощным способом организации информационной инфраструктуры. Использование виртуальных машин позволяет значительно упростить управление и обслуживание серверов, сократить расходы на аппаратное обеспечение, повысить отказоустойчивость и гибкость в администрировании сетевых ресурсов. Применение виртуальных машин в учебном учреждении обеспечивает возможность создания отдельных изолированных сред для различных задач и пользователей, обеспечивая безопасность данных и доступ к ресурсам в соответствии с ролевыми полномочиями.

В целом, основными преимуществами такого подхода можно назвать:

- увеличение коэффициента использования аппаратного обеспечения;
- уменьшение затрат на замену аппаратного обеспечения;

- повышение гибкости использования виртуальных серверов;
- обеспечение высокой доступности;
- повышение управляемости серверной инфраструктуры;
- экономия на обслуживающем персонале;
- экономия на электроэнергии [1].

До ухода продуктов VMware с российского рынка решение vSphere было наиболее распространённым [2]. Многие организации, в частности вузы, сталкиваются с проблемами при переходе на Open source решения, так как не могут найти комплексный аналог экосистемы VMware, наименее заменимой частью которой можно назвать платформу управления рабочими столами VMware Horizon.

Принцип работы VMware Horizon состоит в предоставлении виртуальных рабочих столов и приложений для конечных пользователей через централизованный виртуализированный инфраструктурный пул ресурсов. Эта технология обеспечивает гибкость, масштабируемость и безопасность в предоставлении доступа к рабочим столам и приложениям с любого устройства и из любой точки, основываясь на концепции виртуализации рабочего места.

VMware Horizon работает на базе виртуальной машины (VM), запуская гостевую ОС и приложения в изолированной среде на сервере виртуализации. Пользователи могут подключаться к виртуальному рабочему столу или приложению через клиентский софт или веб-браузер, где все вычислительные операции выполняются на сервере, а результаты передаются на устройство пользователя.

VMware Horizon обеспечивает возможность управления доступом, мониторинга, присвоения ресурсов и обновлений ПО через централизованный интерфейс администрирования. Это позволяет администраторам эффективно управлять бесперебойным функционированием виртуализированной инфраструктуры, гарантируя безопасность и надёжность предоставляемых сервисов для конечных пользователей.

При многочисленных достоинствах данного решения, существенная часть функционала не является необходимой для работы в небольших компаниях и учебных заведениях. Перегруженный интерфейс усложняет работу и увеличивает порог входа в администрирование платформ.

Одним из самых распространённых решений для импортозамещения в области гипервизоров можно назвать proxmox VE (Virtual

Environment) — это платформа виртуализации, предоставляющая интеграцию виртуализации на уровне гипервизора (KVM) и контейнерной виртуализации (LXC) на одном сервере.

Proxmox позволяет администраторам создавать, управлять и мониторить виртуальные машины (VM) и контейнеры (LXC) с помощью веб-интерфейса, обеспечивая гибкость и удобство в управлении виртуализированными окружениями. Помимо этого, он интегрирует в себя гипервизор KVM с удобным управлением, что позволяет легко создавать, управлять, мигрировать и резервировать виртуальные машины, а также имеет удобное API.

Proxmox поддерживает контейнерную виртуализацию с помощью технологии LXC, что позволяет запускать несколько изолированных контейнеров на одном хост-сервере. Это гипервизор поддерживает создание кластеров из нескольких узлов, что позволяет повысить масштабируемость и отказоустойчивость виртуализированных окружений.

Очень важным фактором в пользу выбора Proxmox является активное сообщество пользователей и разработчиков, предоставляющее помощь, обновления и дополнительные ресурсы для эффективного использования платформы. Однако в Proxmox нет никакого встроенного аналога Horizon, а существующие альтернативы, например, OpenUDS, настраиваются и администрируются ещё сложнее проприетарного решения.

Данная проблема решается путём написания собственного аналога, основной идеей которого будет простота развёртывания и использования. В условиях учебного заведения основными функциями платформы управления пулами виртуальных машин является возможность создания такого пула на базе золотого образа, его обновление, остановка и удаление. Для решения поставленных задач была написана программа на языке Python, обращающаяся к API Proxmox, а также удобный и понятный веб-интерфейс.

Литература

- [1] *Крепков Игорь Михайлович, Хорьков Сергей Николаевич.* Виртуализация вычислительных ресурсов в научных исследованиях и учебном процессе университета / Открытое образование. 2012. № 6. <https://cyberleninka.ru/article/n/>

virtualizatsiya-vychislitelnyh-resursov-v-nauchnyh-issledovaniyah-i-uchebnom-protsesse-universiteta (дата обращения: 06.06.2024).

- [2] Сусов Михаил Александрович, Наташкина Елена Анатольевна Особенности внедрения систем виртуализации в Российской Федерации / Евразийская интеграция: современные тренды и перспективные направления. 2023. <https://cyberleninka.ru/article/n/osobennosti-vnedreniya-sistem-virtualizatsii-v-rossiyskoy-federatsii> (дата обращения: 06.06.2024).

Андрей Сурганов

Санкт-Петербург, Центр детско-юношеского технического творчества и информационных технологий Пушкинского района Санкт-Петербурга

ВиЛКа — Виртуальная Лаборатория Кактукс

Ключевые слова: *виртуализация, стенд, kvm*.

Успешное обучение дисциплине «Сетевое и системное администрирование» часто упирается в ограничения инфраструктуры. Предоставить обучающемуся изолированную сеть, разбитую на подсети с несколькими серверами и рабочими станциями, образовательному учреждению крайне затруднительно. Задача усложняется, когда группе обучающихся нужно предоставить несколько подобных сетей. Задача становится практически нереализуемой, когда эти сети нужно поддерживать в актуальном состоянии в течение учебного года. Кроме этого, такой подход лишь с большими трудом подойдёт для проведения итоговых соревнований и олимпиад.

На выручку приходит виртуализация. Здесь есть всё необходимое — сети, устройства различных конфигураций и архитектур, откат к начальному состоянию (снапшоты) и гибкая настройка.

Для организации обучения сетевому и системному администрированию в нашем образовательном учреждении было изучено большое количество решений для управления виртуализацией, но готовые решения с необходимым функционалом так и не были найдены.

Необходимый функционал для педагога:

- создание типовых конфигураций (эталонных стендов) и их клонирование;
- управление стендами;

- контроль стендов во время работы;
- изоляция;
- целостность стенда (запрет пользователю менять принципиально важные настройки);
- удалённый доступ (опционально).

Требования для обучающихся:

- простота использования.

Было принято решение создать собственный инструментарий на базе технологий `Qemu/Kvm` и `NoVnc`. Проект получил название «ВиЛ-Ка» — Виртуальная Лаборатория Кактус.

В качестве бэкенда выступает(ют) сервер(ы) виртуализации. Помимо виртуальных машин на нём работает служба `NoVNC`, которая предоставляет возможность подключаться к экранам виртуальных устройств по протоколу `HTTP`. Фронтенд работает под управлением `Nginx`. `Nginx` занимается авторизацией пользователя, предоставляет панель управления и проксирует пользователя на сервер(ы) виртуализации, к `NoVNC`, к экранам виртуальных устройств.

Как это работает: клиент с помощью веб-браузера подключается к веб-серверу, авторизуется на нём и получает доступ к панели управления своим стендом. В панели управления пользователю доступна сама панель с элементами управления и топология с устройствами-ссылками на виртуальные устройства стенда.

Помимо этого, для администрирования стендов создан инструментарий со следующим функционалом:

- получение команд от панели управления (проверка на вменяемость и выполнение);
- клонирование эталонных стендов;
- создание сценариев запуска `NoVNC` (для каждого стенда отдельный сценарий);
- создание конфигурационных файлов `Nginx` (каждый стенд - это отдельный виртуальный хост);
- создание панели управления для каждой группы однотипных стендов;
- набор служебных скриптов.

В итоге получился продукт, готовый к использованию для обучения и соревнований и олимпиад, в том числе онлайн. Уже 3 года на нём проводятся открытые детско-юношеские соревнования по Linux «СасTUX» и межрегиональный конкурс Linux-Skills в рамках фестиваля технического творчества «Техноактус».

Пётр Леляев

Москва, ГАУ «Центр цифровизации образования»

Проект: Управление компьютерным кабинетом

<https://hub.mos.ru/mos/src/comp-cab-setup>

Система управления компьютерным классом для МОС

Аннотация

В данной статье приведён обзор системы управления компьютерным классом, созданной в ГАУ «Центр цифровизации образования». Рассмотрены её основные отличия и преимущества по сравнению с используемыми на проприетарных ОС решениями, в частности, Veyon.

Ключевые слова: *МОС, ssh, управление, кабинет, информатика.*

Активное развитие информационных, и, в частности, сетевых технологий не обошло стороной образовательный процесс. В первую очередь это касается уроков информатики, как наиболее технически оснащённого процесса в школьном образовании. Достаточно продвинутые учителя при работе на проприетарных ОС использовали в работе различные инструменты, в первую очередь Veyon. При переходе на свободные ОС возможность пользоваться Veyon сохранилась (в частности, на Astra Linux [1] и ОС «Альт» [2]). Однако главная проблема при использовании Veyon в образовательном процессе — сложность его настройки. Значительная часть учителей и даже администраторов не может настроить его самостоятельно. В то же время постоянный контроль за рабочими местами учеников может не являться необходимым, но несколько инструментов, такие как обмен файлами, запрет определённых действий учеников, удалённый вход и выход из системы, очевидно нужны подавляющему большинству учителей. Возникла необходимость в создании инструмента, где данные функции будут работать «из коробки», без необходимости выполнения команд в консоли. Таким инструментом стала программа «Управление

компьютерным кабинетом», созданная в ГАУ «Центр цифровизации образования».

Эта программа изначально была рассчитана не как аналог, а как дополнение к Veyon, и связывает компьютер учителя и учеников по протоколу ssh. Главным её преимуществом является то, что достаточно в графическом интерфейсе внести имена и адреса хостов с логином и паролем администратора, после чего доступ по ssh настраивается в автоматическом режиме. На рис. 1 и 2 представлен вид текущей версии программы. Она предоставляет различный функционал для учителя и администратора для случая, когда это разные люди и они не вмешиваются в работу друг друга.

Возможности, доступные для учителя:

- раздача, сбор и очистка заданий со всех выбранных хостов;
- архивация и восстановление домашней папки на хостах;
- отображение имени хоста, смена обоев, открытие браузера, удалённый вход и выход из системы;
- открытие на сервере проводника с подключённой папкой на одном из хостов;
- также учитель может работать и переключаться между профилями — это файлы, хранящие информацию о списке хостов вместе с логинами и паролями. У учителя есть возможность организации одновременного выхода из системы на всех хостах и централизованного входа в систему под другим логином, что может быть полезно, когда в одном кабинете занимаются несколько разных классов.

Возможности программы для администратора:

- проверка доступности хостов, настройка доступа по ssh, получение IP по MAC-адресу;
- создание и удаление учётных записей, включение и выключение автологина;
- установка и настройка Veyon на сервере и хостах, копирование и удаление ключей;
- выполнение команд на хостах, установка и удаление программ;
- ограничения профиля — запрет на изменение рабочего стола, интернет, подключение к wi-fi сетям, устройства ввода, а также защита паролем самой вкладки настройки;

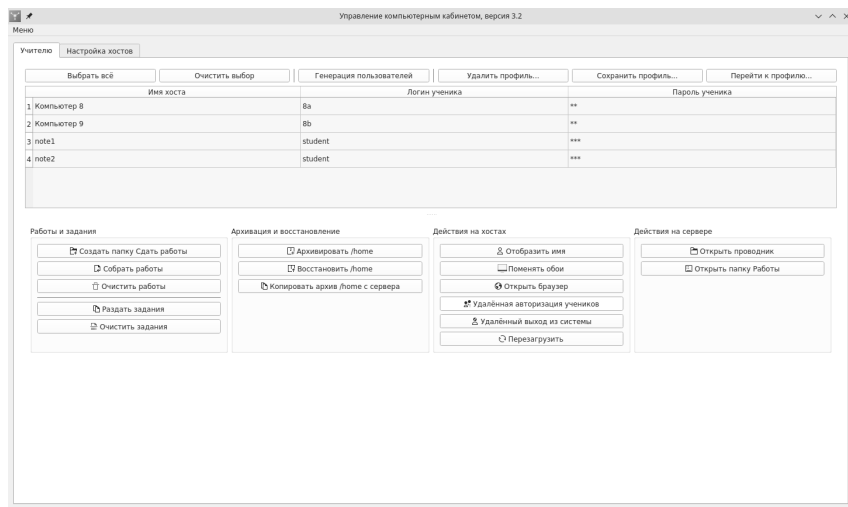


Рис. 1: Управление компьютерным кабинетом, вкладка учителя

- простая генерация, импорт и экспорт адресов хостов, логинов и паролей администраторов и учеников.

Отдельно следует упомянуть пересекающиеся с Veuon возможности, например, удалённый вход в систему. Разработчиков не устроила реализация входа в Veuon через xdotool, но других вариантов в Linux просто не существует, потому что удалённый вход в систему был реализован через доработку sddm, чтобы можно было отправить специальным образом сформированную команду для логина. Такое решение невозможно предложить в апстрим Veuon, поскольку оно зависит от sddm.

Программа создавалась и развивается в тесном сотрудничестве с фокус-группой сетевых администраторов московских школ, в настоящее время она входит в репозиторий дистрибутивов систем МОС и РОСА. Для программы существует подробная инструкция [3], в частности, описывающая решения часто возникающих проблем. Кроме компьютерного класса, система управления может также использоваться для любых устройств под управлением МОС. Разработчики

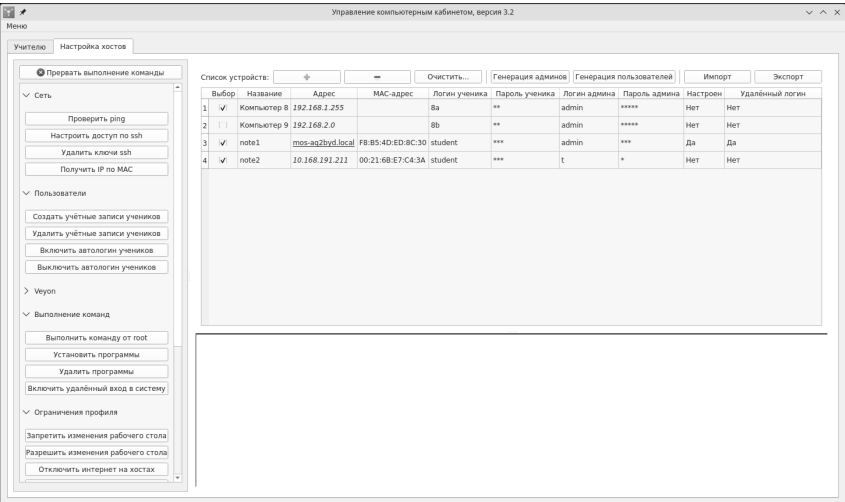


Рис. 2: Управление компьютерным кабинетом, вкладка настройки хостов

рассчитывают на более широкое внедрение данной программы в образовательный процесс.

Литература

[1] Информатизация в школе, Veyon — установка и настройка в Astra Linux Common Edition, 2021

[2] Базальт СПО, Как настроить программу для управления компьютерным классом Veyon, 2024, <https://habr.com/ru/companies/basealtspo/articles/799727/>

[3] Центр цифровизации образования, Управление компьютерным классом, 2024, <https://it-help-school.ru/>

Сергей Быковский, Ирина Степанова, Рустам Халабия
Москва, МИРЭА — Российский технологический университет, Московский
технический университет связи и информатики (МТУСИ)

Опыт применения свободного программного обеспечения для организации серверов удалённой загрузки в образовательной организации

Аннотация

Перевод процесса обучения студентов на свободное программное обеспечение сталкивается с различными трудностями, в том числе связанными с необходимостью запуска на операционных системах Linux программ, разработанных для других операционных систем. Авторами было осуществлено развёртывание учебного компьютерного класса в режиме «толстого клиента» с загрузкой операционной системы из образа по сети. Описываются особенности построения лабораторного класса с сетевой загрузкой гетерогенных клиентов, решение некоторых задач, возникающих при запуске на клиентах приложений в wine среде. Проводится анализ сложностей, возникающих при аутентификации на клиентских машинах.

Ключевые слова: *свободное программное обеспечение, толстые клиенты, ОС Linux, PXE-образы, wine.*

С переходом на свободное программное обеспечение в условиях учебного процесса стали возникать определённые сложности с управлением рабочими местами студентов в компьютерных классах, а также с аттестацией и проверкой навыков и умений учащихся. Это происходит, когда в течение дня приходится часто переключаться с одного программного продукта на другой, либо когда каждый студент выполняет работы на программных продуктах по своему усмотрению, например, курсы, связанные с программированием с интегрированием различных компиляторов и интерпретаторов.

При однородном «парке» машин, одним из вариантов является установка и полная настройка необходимого программного обеспечения на одной машине, а затем её «клонирование» [1]. При наличии высокоскоростной ЛВС более выгодным может стать режим «толстого клиента». В качестве клиентских ОС были выбраны AstraLinux, Ubuntu и «Alt».

Создание PXE-образа толстого клиента и его запуск в точном соответствии с [2][3] выявили следующую неприятную особенность —

Linux-ПО на клиентах запускалось за адекватное время, старт wine-программ выполнялся недопустимо долго — порядка полутора минут.

Пересборка образа для работы в режиме «Тонкий клиент» сократила время запуска до 15 секунд, но при подключении к серверу более трёх клиентов произошло сильное падение производительности.

«Полностью ручное» создание образов через стандартный для Debian-систем `debootstrap` завершилось у авторов неудачей из-за проблем с драйверами сетевой карты. От использования LTSP-инструментария пришлось отказаться из-за сетевой политики безопасности (частичной блокировки DHCP-трафика).

В итоге, сам образ был сформирован «родным» пакетом, а его «обвязка» написана вручную. Настройка обвязки потребовала ручной правки `grub.conf`, фрагмент которого принял вид:

```
menuentry 'Ubuntu 22.04' {
  set i_name="Jeley-Fish"
  echo "Loading $i_name kernel ..."

  linux /jeley-fish/vmlinuzinitrd=ltsp.imginitrd=initrd.img
    root=/dev/nfsnfsroot=192.168.1.20:/srv/tftp/jeley-fish ltsp.image=jeley.img

  echo "Loading initramfs, Please, wait..."
  initrd/jeley-fish/ltsp.img /jeley-fish/initrd.img
}
```

При работе в полученной системе для студентов наиболее неприятным моментом оказался факт, что все окна программ, запускаемых через wine, теряют статус модальных, из-за чего складывается ложное впечатление о зависании программы. Эта особенность наблюдается как для Astra с графической средой Fly, так и для Ubuntu со средой Gnome.

Для администратора особенность работы связана с wine. Wine создан его разработчиками таким образом, что для каждого пользователя Linux-системы он создаёт профиль (при необходимости и несколько профилей) в его домашнем каталоге. Создание для каждого студента отдельной учётной записи очень быстро истощает дисковое пространство сервера, поэтому мы ограничились одной учётной записью для всех студентов.

Ещё одна особенность связана с процессом аутентификации, точнее, с особенностями работы с паролями в Ubuntu и Astra. Просмотр shadow-файлов показал различную длину хэшей и использование разных алгоритмов хеширования с постоянной солью у Ubuntu и динамически изменяемой для Astra. Войти в ОС на клиенте можно только

если сервер работает под управлением одноимённой ОС. Ещё одна сложность ожидается при добавлении к набору образов клиентов на основе ОС «Альт», в котором в `/etc/shadow` записаны строки только системных учётных записей, а для пользовательских аккаунтов создаются индивидуальные shadow-файлы в `/etc/tcb`.

Возможно, будет использован сервер аутентификации, подобно описанному в [4].

Перспективы дальнейшего развития проекта:

- выполнить пересборку `initramfs`-файлов и сократить их цепочку до одного;
- обеспечить единообразие хэширования паролей;
- создать единый `/home`-раздел при работе с любым из образов;
- организовать единый каталог `wine`-профилей для всех пользователей;
- интегрировать данное решение в образовательную среду с вариантами обучения и оценки знаний.

Литература

- [1] Алексеев С. П., Михайлова А. О., Оптимизация администрирования операционной системы Astra Linux при переходе подразделений ОВД на отечественное программное обеспечение // Охрана, безопасность, связь. 2021. №6–2. С.15–21. <https://www.elibrary.ru/item.asp?id=46333684>.
- [2] Подготовка инфраструктуры PXE на Astra Linux. <https://wiki.astralinux.ru/pages/viewpage.action?pageId=263031254>.
- [3] Терминальный сервер LTSP (ltsp-server-standalone) на базе Astra Linux. <https://wiki.astralinux.ru/pages/viewpage.action?pageId=53642256>
- [4] Даньшина А. В., Докин А. Д., Ковцур М. М., Разработка сервера аутентификации на базе операционной системы Astra Linux // Региональная информатика и информационная безопасность. 2020. С.262–265. <https://www.elibrary.ru/item.asp?id=48027341>.

Павел Макаров

Сыктывкар, ФМИ ФИЦ Коми НЦ УрО РАН, СЛИ

<https://makarovpa.ru>

Практика применения свободного программного обеспечения в преподавании дисциплин физико-математической и технической направленности

Аннотация

Данная статья представляет собой результат обобщения опыта преподавания различных дисциплин физико-математической и технической направленности студентам двух вузов города Сыктывкара: Сыктывкарского государственного университета имени Питирима Сорокина (СыктГУ) и Сыктывкарского лесного института (СЛИ) на протяжении 2011–2024 г.

Обсуждаются преимущества использования свободного программного обеспечения в данной области деятельности, а также описываются некоторые решения и методики, разработанные автором.

Ключевые слова: *образование, наука, свободное программное обеспечение.*

Введение

Использование свободного программного обеспечения (СПО) в образовании вообще и в высшей школе в частности не является в последнее время чем-то уникальным. В качестве примеров можно привести множество работ [1, 2, 3, 4, 5, 6, 7, 8] в которых описан опыт применения СПО при обучении студентов.

Таким же образом изначально развивалась и данная работа. Первыми шагами на этом пути было решение автора при преподавании студентам СыктГУ направлений «Физика» и «Радиофизика» в 2011–2012 учебном году дисциплины «Программирование на Си» использовать СПО вместо повсеместно распространённых проприетарных сред разработки.

При этом, естественно, возник вопрос доступности выбранного инструментария в компьютерных классах СыктГУ. Решением данной проблемы на начальном этапе было выбрано свободное распространение в компьютерной сети СыктГУ образа виртуальной машины с

предварительно установленной ОС Debian GNU/Linux и всем необходимым начинающему программисту инструментарием (`vim`, `gcc`, `glibc`, `gdb`, `make`).

Единственный недостаток этого решения состоял в том, что его внедрение повышало «порог вхождения», т.е. прежде чем, собственно, приступить к выполнению лабораторных работ по дисциплине «Программирование на Си», требовалось сначала привить студентам навыки использования эмуляторов виртуальных машин, обучить их основам командной строки UNIX-like ОС, а также принципам работы с конкретным системным и прикладным ПО.

Глубокое осмысление указанной выше проблемы показало, что такие усилия для преодоления достаточно высокого «порога вхождения» не очень оправданы при изучении единственной дисциплины учебного плана. Более того, в отрыве от остальных курсов (которые традиционно преподаются с применением проприетарных средств) такая практика может создать у студентов впечатление искусственности и бесполезности происходящего. А это — эффект прямо противоположный изначально задуманному.

Процесс преодоления этого противоречия стал следующим этапом развития проекта. В его ходе был накоплен достаточно большой демонстрационный и методический опыт, убедительно доказывающий, что полученные навыки и знания могут быть эффективно использованы при освоении самых различных предметов. Важно подчеркнуть, что этими дисциплинами являются не только компьютерные науки, но и математические, общепрофессиональные и специальные курсы.

Таким образом, в смысле широты охвата содержание данной работы близко к материалам, описанным в источниках [9, 10, 11, 12], но существенно отличается как по фактическому наполнению, так и по используемым техническим решениям.

Текущее состояние проекта

Таким образом, на текущий момент проект представляет собой сервер, работающий под управлением ОС Ubuntu и доступный для удалённой работы студентов в любое время и из любой точки. Для доступа к серверу может использоваться любое устройство с установленным браузером и доступом к сети Интернет. Технически это реализовано с помощью `Shell-in-A-Box`. Также поддерживается доступ к серверу по протоколу SSH.

При этом в работе с сервером активно используются следующие особенности.

1. Скрипты, автоматизирующие добавление и удаление пользователей системы, согласно спискам академических групп.
2. Web-сервер (**Apache**), применяемый для методического обеспечения работы студентов.
3. Установлен набор разработчика: интерпретатор **Python**, трансляторы языков **C** и **C++** (**gcc** и **g++**), ассемблеры (**gas**, **nasm**), стандартная библиотека **Си** **glibc**, библиотека **MathGL**, система сборки **make**, отладчик **gdb**, система контроля версий **git**.
4. Для выполнения качественной графики установлены **gnuplot** и **Asymptote**.
5. Установлены системы компьютерной математики **Maxima** и **Octave**.
6. В качестве сред для обработки статистических данных используются **R** и **ROOT**.
7. Для подготовки отчётов установлен **TeXLive**.
8. Написано прикладное ПО, позволяющее проводить тестирование студентов по тем или иным курсам прямо на сервере.
9. Вся деятельность пользователей протоколируется, автоматически ведутся журналы сеансов работы.

Заключение

Подводя итоги, можно сказать, что описываемый в рамках данной работы подход был широко апробирован автором при преподавании самых разнообразных предметов студентам бакалавриата и магистратуры пяти направлений: «Физика», «Радиофизика», «Математика и компьютерные науки», «Прикладная математика и информатика» и «Информационные системы и технологии».

К числу охваченных таким образом дисциплин можно отнести следующие: «Программирование на **Си**», «Технология программирования», «Объектно-ориентированное программирование», «Теоретическая механика», «Квантовая механика», «Численные методы решения волновых уравнений», «Учебная практика», а также курсовое проектирование, научная работа студентов и выполнение ими выпускных квалификационных работ.

Таким образом, работу данного проекта можно считать достаточно успешной. К преимуществам разработанной методики относятся:

- Независимость от используемого в вузе аппаратного и программного обеспечения.
- Доступность системы.
- Универсальность подхода при освоении различных дисциплин.
- Удобство работы как студентов, так и преподавателя.
- Гибкость в настройке и использовании.

Литература

- [1] *Иванова Н. Ю.* Опыт использования OpenOffice.org в курсе «Программное обеспечение ЭВМ» на математическом факультете МПГУ // Четвёртая конференция «Свободное программное обеспечение в высшей школе»: тезисы докладов. Переславль, 30 января — 1 февраля 2009. с. 69–71.
- [2] *Зайцев И. В.* О преподавании курса по алгоритмизации на основе языка JavaScript и открытого ПО // Четвёртая конференция «Свободное программное обеспечение в высшей школе»: тезисы докладов. Переславль, 30 января — 1 февраля 2009. с. 96–98.
- [3] *Лагунов А. Ю.* Выбор среды разработки для обучения студентов программированию на языке JAVA // Четвёртая конференция «Свободное программное обеспечение в высшей школе»: тезисы докладов. Переславль, 30 января — 1 февраля 2009. с. 99–101.
- [4] *Юмова К. Л.* Об одном подходе к преподаванию дисциплины «Информатика» при переходе на свободное программное обеспечение // Вестник Бурятского государственного университета. 2013. Вып. 15. с. 70–73.
- [5] *Алексеев Е. Р., Мандрыкина Д. С.* Использование свободных компиляторов при изучении технологий параллельного программирования // Объединённая конференция «СПО: от обучения до разработки»: материалы конференции. Переславль-Залесский, 19–22 мая 2022. с. 56–60.
- [6] *Алексеев Е. Р., Гончаров С. В.* Свободные библиотеки интервальных вычислений при подготовке бакалавров и магистров направления «Математика и компьютерные науки» в Кубанском государственном университете // Объединённая конференция «СПО: от обучения до разработки»: материалы конференции. Переславль-Залесский, 19–22 мая 2022. с. 60–65.

- [7] *Балашов В., Курячий Г.* Как мы добывали огонь. Организация аудиторного практикума по курсу «ЯП Python» на останках дистанционного обучения // Восемнадцатая конференция «Свободное программное обеспечение в высшей школе»: материалы конференции. Переславль-Залесский, 27–29 января 2023. с. 60–62.
- [8] *Смирновский А.* Опыт использования СПО OpenFOAM для обучения основам вычислительной гидродинамики в СПбПУ // Восемнадцатая конференция «Свободное программное обеспечение в высшей школе»: материалы конференции. Переславль-Залесский, 27–29 января 2023. с. 31–33.
- [9] *Чичкарев Е. А., Чичкарев К. Е.* Интегрированный пакет математических расчётов S.A.G.E.: использование в преподавании // Четвёртая конференция «Свободное программное обеспечение в высшей школе»: тезисы докладов. Переславль, 30 января — 1 февраля 2009. с. 79–82.
- [10] *Алексеев Е. Р., Грушевский С. П.* Опыт использования свободного программного обеспечения при подготовке учителей математики и информатики в Кубанском Государственном Университете // Объединённая конференция «СПО: от обучения до разработки»: материалы конференции. Переславль-Залесский, 19–22 мая 2022. с. 30–34.
- [11] *Волканов Д. Ю., Курячий Г. В.* Цепочка кафедральных курсов, использующих свободное программное обеспечение // Объединённая конференция «СПО: от обучения до разработки»: материалы конференции. Переславль-Залесский, 19–22 мая 2022. с. 65–67.
- [12] *Панюкова А. А.* Применение СПО в серии дисциплин основного, общего гуманитарного и социально-экономического цикла для IT-специалистов // Объединённая конференция «СПО: от обучения до разработки»: материалы конференции. Переславль-Залесский, 19–22 мая 2022. с. 73–75.

Андрей Васильев

Ярославль, Ярославский государственный университет им. П. Г. Демидова

Переход компьютерного класса 7-го учебного корпуса ЯрГУ на ОС Simply Linux

Аннотация

В докладе рассматривается история использования ОС семейства Linux на компьютерах в учебном компьютерном классе университета. Обсуждается выбор ОС Simply Linux как удовлетворяющей ключевым критериям самостоятельности, открытости внутреннего процесса разработки и доступности для коммерческого использования студентами после выпуска. Рассматриваются вопросы перехода с Debian на дистрибутив Simply Linux и настройки системы для использования в учебном процессе.

Ключевые слова: *Debian, Simply Linux, администрирование.*

Компьютерный класс

В учебных программах факультета информатики и вычислительной техники с конца 2000-х годов существует курс «Операционные системы семейства UNIX и их администрирование». В его рамках студенты учатся работать в командном интерфейсе, автоматизировать действия с использованием Bash и администрировать ОС GNU/Linux.

Первоначально Mandriva Linux был установлен в качестве альтернативы Windows в компьютерном классе на двенадцать машин. В начале 2010-х был осуществлён переход на дистрибутив Debian, который впоследствии стал единственной ОС. В 2022 встал вопрос о переходе на отечественное ПО.

Выбор дистрибутива

При выборе дистрибутива на замену Debian руководствовались следующими ключевыми критериями:

- Решение должно быть самодостаточным, не являться прослойкой для получения доступа к Debian.
- У выпускника должна быть возможность применения собственных навыков без обязательств оплаты, но с возможностью за доплату получить нужный уровень поддержки.

- Процесс разработки базового ПО должен быть открытым, чтобы студенты могли в нём участвовать и повышать квалификацию.

Данным критериям удовлетворяют как Debian, так и отечественные разработки: Sisyphus, РОСА «Фреш» и Calculate Linux. Положительным фактором выбора решения Simply Linux стала длительная история взаимодействия компании «Базальт СПО» с образовательными учреждениями и понимание их нужд.

Особенности перехода на Simply Linux

В результате выполнения перехода можно отметить следующие различия между дистрибутивами Debian и Simply Linux.

Документация. Для *Debian* в лёгком доступе находится количество статей и руководств по выполнению почти любых задач. Для *Simply Linux* информацию приходится добывать путём изучения официальной документации и ALT Linux Wiki.

Доступность ПО в репозитории. Репозиторий *Debian* считается одним из самых больших среди Linux-дистрибутивов. В репозитории *Simply Linux* были найдены все необходимые приложения или их альтернативы.

Установка ПО не из репозитория. Для *Debian* разработчики ПО часто самостоятельно создают пакеты. Для *Simply Linux* такого уровня поддержки ПО нет. Решением проблемы становятся альтернативные системы доставки приложений: Flatpak, Docker, AppImage и бинарные сборки.

Скорость работы пакетного менеджера. В *Debian* пакеты зачастую включают в себя скрипты, которые выполняются после установки пакетов. Пакеты *Simply Linux* не включают скрипты, что значительно ускоряет процесс настройки системы.

Удобство пакетного менеджера. АРТ, применяемый в *Debian*, сильно эволюционировал, стал предоставлять удобное единое приложение `apt`. В *Simply Linux* для получения схожего уровня комфорта работы был использован инструментарий EPM [1].

Безопасность системы по умолчанию. По сравнению с *Debian* в *Simply Linux* по умолчанию закрыт доступ к ряду системных конфигурационных файлов, реализован более строгий контроль над действиями пользователя, который настраивается с помощью инструментария `control` [2].

Настройка ПО и его интеграция

Компьютерный класс используется в основном для проведения практических занятий по учебным курсам, связанным с информационными технологиями. При настройке системы уделялось внимание скорости запуска самой системы и целевого ПО.

Компьютеры не интегрированы в домен университета, так как интеграция требует значимых ресурсов, но выигрыш от неё незначителен. Внутри сети университета студентам доступен только сетевой диск, а другие сетевые сервисы не интегрированы в единую систему входа. Доступ к сетевому диску организован через GVFS-слой.

Ключевыми средами для работы являются кросс-платформенные бесплатные инструменты от компании JetBrains. Они устанавливаются путём распаковки архивов с подготовленными разработчиками бинарными файлами. Ряд других учебных приложений установлен через систему дистрибуции приложений Flatpak. Обычные пользователи могут их запускать, но не могут модифицировать благодаря правилам Polkit [3].

Для ускорения запуска сред разработки была проведена работа по выделению их кэша из домашнего каталога в отдельные архивы. Пользователь вместо среды разработки запускает скрипт-обёртку, который разворачивает архив и затем запускает приложение. Таким образом размер профиля был значительно снижен, но эффективность использования кэша сохранена.

Литература

- [1] Единая команда управления пакетами EPM <https://wiki.etersoft.ru/Epm>
- [2] Alt Linux Wiki control <https://www.altlinux.org/Control>
- [3] Шаблон правил polkit для Flatpak <https://github.com/flatpak/flatpak/blob/main/system-helper/org.freedesktop.Flatpak.rules.in>

Екатерина Прокофьева, Леонид Чашкин, Кирилл Букин
Москва, Национальный исследовательский университет «Высшая школа экономики», департамент компьютерной инженерии, Московский институт электроники и математики имени А. Н. Тихонова

Alt Linux как среда решения образовательных, инженерных и научных задач

Аннотация

Одной из острых проблем, которые мешают широкому распространению СПО в образовании, является нехватка методической литературы. Эта проблема решается на базе курсов ряда университетов, в том числе на базе дисциплины «Операционные системы» и индивидуальных практик курса по ОП ИВТ МИЭМ НИУ ВШЭ. Дополнительно создаётся общеуниверситетский факультатив для более широкого распространения прикладных практик по дисциплине. С точки зрения «Базальт СПО», проблема и задачи могут быть решены на базе ресурса «Библиотека Alt Linux» — здесь имеется общеобразовательная и общенаучная литература, содержащая элементы методик обучения с использованием свободных программных сред Maxima, Octave, Scilab, языкам программирования Free Pascal, Python, C++ и др. Среди других проблем, которые предстоит преодолеть — предубеждение и скепсис привыкших пользоваться зарубежным проприетарным ПО, и нехватка заинтересованных в развитии СПО.

В условиях реализации стратегии импортозамещения показать обучающимся возможности и преимущества отечественных операционных систем на примере ОС «Альт» (ранее Alt Linux) в решении актуальных прикладных инженерных, статистических и исследовательских задач: запуск процессов в рамках базового функционала, разграничения прав, работы с репозиториями, установки библиотек и пакетов, функционирования образовательных, программных, статистических, графических, сетевых, ГИС и специализированных управленческих приложений. Актуальные тематические направления. Основы администрирования линукс-подобных ОС. Основы работа с геопространственными данными и моделями. Основы программирования и работы с кодом в управлении процессами.

Ключевые слова: *Alt Linux, факультативы, инструменты решения, модели, автоматизация администрирования.*

Введение

Программные продукты класса Enterprise нередко вырастают из образовательных проектов. Ранее студенты в предшествующих проектах разработали систему анализа геопространственных данных и построения карты рельефа. В своей работе они использовали свободную геоинформационную систему QGIS, включённую в дистрибутив ОС «Альт Образование». За предшествующие три года был разработан базовый пул практик по курсу ОС с учётом особенностей 9 версии «Альт Образование» (требуется полное обновление с учётом особенностей 10 версии, а также дифференциации дистрибутивов), создана пробная программа тренажёр на базе «Альт» для курса ВСиКС, создано несколько обучающих видеороликов. Требуется привлечение студентов к развитию инструментов для профессионального обучения, аналитики и исследований на базе платформ «Альт». Работа в проектах СПО отлично помогает реализовать идеи, навыки и опыт студентов.

СПО активно используется и в научных проектах НИУ ВШЭ помимо области работы с геопространственными данными, в частности проводят классификацию опухолевых клеток с использованием моделей машинного обучения в среде ОС «Альт». В Брестском государственном техническом университете СПО используют для ЭЭГ-мониторинга психического состояния пользователя. Поэтому на базе мощностей МИЭМ НИУ ВШЭ и Базальт СПО можно продолжить развитие потенциала профессиональной учебно-методической деятельности и академической работы, разработать инструменты к эффективной реализации. Также есть потенциал для сотрудничества с другими университетами и НИИ.

Примеры решений

Один из примеров. На базе СПО были построены актуальные цифровые модели рельефа местности с использованием модулей QGIS.

На базе собранных открытых пространственных данных местности была произведена обработка. В программе QGIS доступно наложение этих данных на карту в виде теневого рельефа, что позволяет создать эффект объёмной карты. Далее был наложен на карту цветовой градиент по высоте рельефа. Была использована возможность добавить изолинии на готовую карту рельефа.

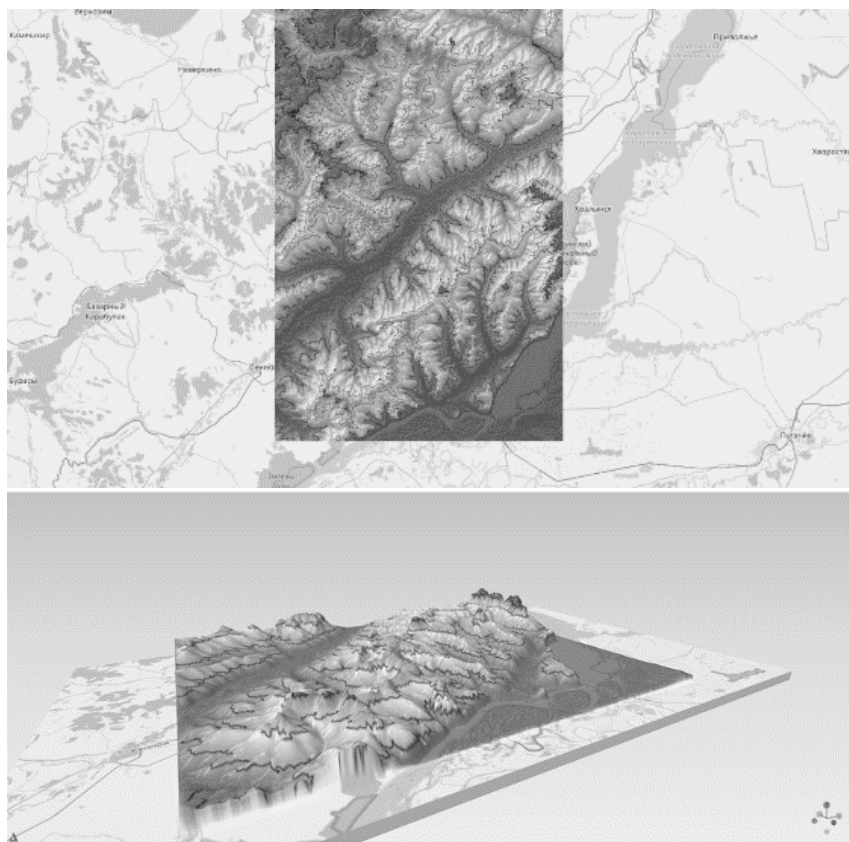


Рис. 1: Карта и Цифровая модель Вольского карьера

Для построения цифровой модели был использован модуль **qgis2-threejs**, позволяющий перевести полученную карту рельефа в 3D-модель с различными настройками сжатия/растяжения по высоте и ширине. Разработанный алгоритм был использован для построения карты рельефа территории Вольского карьера в Саратовской области с использованием операционной системы «Альт» и предустановленной программы QGIS. Результаты представлены на рис. 1.

Ещё одним примером работы с данными на базе ОС «Альт» выступает широкое использование программных модулей и приложений для вывода разноформатных данных и решений. В частности, геолокаций на базе модулей Python, построение рекреационных зон [1–4]. Один из примеров такого решения для обучающихся представлен во фрагменте кода разметки объектов и на рисунках 2–3.

Фрагмент кода разметки объектов локаций:

```
# Сохранение карты в html
m1.save('map1.html')

m2 = folium.Map(# создание маркеров и меток рекреационных зон с пояснительным контекстом
    location=[55.63,37.33],
    tiles='openstreetmap',
    zoom_start=14)

folium.Marker(
    location=[55.63,37.31],
    popup='<i>парк Рассказовка</i>',
    tooltip='Место прогулок и парковая зона').add_to(m2) # Добавляем метку первой ближайшей
                                                            культурно-рекреационной зоны с пояснительным контекстом

folium.Marker(
    location=[55.65,37.33],
    popup='<i>Переделкино</i>',
    tooltip='Место прогулок и культурный объект',
    icon=folium.Icon(color="red")).add_to(m2) # Добавляем метку второй ближайшей культурно-
                                                            рекреационной зоны с пояснительным контекстом

folium.CircleMarker(
    location=[55.63,37.33],
    radius=300,
    popup="Рекреационные зоны метро Рассказовка",
    color="#3185cc",
    ...)
```

На базе работы с терминалом обучающимся демонстрируется структура поточных процессов и работы с нитями как элементы и инструменты базового администрирования операционной системы. Пример написания скрипта для запуска нитей представлен в листинге ниже.

Фрагмент кода для запуска нитей и поточных процессов:

```
int main (int argc, char *argv[])
{
    pthread_t threads[NUM_THREADS];
    int rc;
    long t;
    for(t = 0; t < NUM_THREADS; t++){
        printf("In main: creating thread %ld\n", t);
        rc = pthread_create(&threads[t], NULL, PrintHello, (void *)t);
        pthread_join(threads[t], 0);
        if (rc)
        {
            printf("ERROR: return code from pthread_create() is %d\n", rc);
            exit(-1);
        }
    }
}
```



Рис. 2: Разметка и нанесение пути между объектов локаций.

```
pthread_exit(NULL);
}
```

Также для запуска нитей можно использовать установку пакетов из репозитория, например, `libcthread-20130723-alt1.x86_64.rpm`.

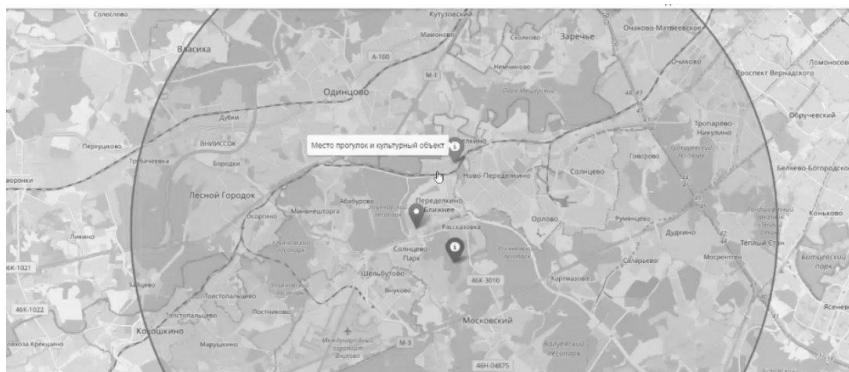


Рис. 3: Нанесение радиуса локационной зоны и меток.

Процессы добавления пользователей в ОС Альт отображены в листинге ниже.

```
# If env_reset is disabled, sudo will NOT reset the environment
# to only contain the fixed list of variables.
# See sudoers(5) for details.
#Defaults:WHEEL_USERS !env_reset

# Preserve DISPLAY and XAUTHORITY environment variables
# for "agrp" group members.
Defaults:XGRP_USERS env_keep += "DISPLAY,XAUTHORITY"

##
## Runas alias specification
##

##
## User privilege specification
##
# root ALL=(ALL:ALL) ALL
mentor ALL=(■)
## Uncomment to allow members of group wheel to execute any command
```

В рамках администрирования, управления и автоматизации процессов студентами было разработано приложение для автоматического добавления больших групп пользователей.

А также успешно реализованы запуски пакетов по экономико-финансовому анализу фондового рынка, созданы приложения в рамках технико-экономического аудита и использовано СПО по разнофункциональной направленности. В частности, для решения задач инженерно-технического, экомониторингового и социального характера, например, для анализа и совершенствования работы с данными в рамках комплексных отчётов по устойчивому развитию предприятий ресурсного сектора и нефтехимической отрасли.

Заключение

Таким образом, на основе СПО и цифровых платформ «Базальт СПО», в МИЭМ НИУ ВШЭ разрабатываются материалы для интерактивных курсов, учебно-методические материалы для последующей публикации пособия и дополнительные учебные программные приложения по факультативу.

Ключевыми задачами выступают:

- Разработка прикладных задач для отработки практических навыков
- Разработка теоретических материалов для самостоятельного изучения
- Изменение программного комплекса для изучения дисциплины

- Изменение комплекта лабораторных работ для более углублённого изучения дисциплины
- Разработка фактуры учебных материалов для общеуниверситетского факультатива и магистров

По результатам изучения формируются навыки: работа с дистрибутивами ОС «Альт», администрирование Linux-подобных систем, работа с пакетами и ядром, работа с Python, C, работа с wiki-гипертекстовой документацией, знание технических и образовательных стандартов, работа в научных и образовательных проектах. Умения: написание образовательных материалов для учебного и профессионального использования, проведение профессиональных практик и мастер-классов по администрированию и функциональной работе с ОС, обучение качественной работе с информационными платформами, в частности с отечественными, в условиях импортозамещения. А по итогу формируется и развивается интерактивная и программно-методическая база приложений, инструментов и актуальных решений.

Литература

- [1] *Vostrikov A. V., Prokofeva E. N., Goncharenko S. N., Gribanov I. V.* Analytical modeling for the modern mining industry // Eurasian Mining. 2019. No.2(32). P.30–35. DOI 10.17580/em.2019.02.07
- [2] *Goncharenko S. N., Duong L. B., Petrov M. V., Stoyanova I. A.* Modeling of parameters of innovation water-protection measures on the basis of industrial-technological indices of coal mining at Vietnam enterprises (2014) *Gornyi Zhurnal*, (9) , pp.143–146.
- [3] *Prokofeva E. N., Vostrikov A. V., Shapovalenko G. N., Alvarez A.* The development of effective geomonitring for mining area with industrial review // Eurasian Mining. 2017. No. 2. P. 61–63.
- [4] *Temkin I., Deryabin S., Konov I.*: Soft computing models in an intelligent open-pit mines transport control system. *Procedia Computer Science*, Vol. 120. 2017.

Леонид Чашкин, Екатерина Маршутина

Москва, НИУ ВШЭ

Проект: проект VPCS

https://github.com/lbchashkin/vpcs/tree/ipv6_gateway

Расширение возможностей проекта VPCS при моделировании IPv6-адресации с использованием программного эмулятора компьютерных сетей GNS₃

Аннотация

Данная работа посвящена развитию проекта VPCS при моделировании компьютерных сетей в программном эмуляторе GNS₃. В работе предлагается решение проблемы ограниченных возможностей виртуальных компьютеров VPCS при моделировании IPv6-адресации в GNS₃. Разработан механизм добавления и использования статического маршрута по умолчанию при IPv6-адресации для виртуальных компьютеров VPCS.

Ключевые слова: *компьютерные сети, IPv6-адресация, моделирование, VPCS, GNS₃.*

В настоящее время огромное значение приобретает использование IPv6-адресации, которая решает проблему недостатка адресов в протоколе IPv4 [1]. В связи с этим важным аспектом во время обучения администрированию компьютерных сетей становится моделирование IPv6-адресации. Тем не менее, на данный момент в этой области существует проблема ограниченности функционала некоторых эмуляторов (например, GNS₃). Целью данной работы является решение проблемы моделирования статической IPv6-адресации в программном эмуляторе GNS₃ с использованием виртуальных компьютеров VPCS.

Среди основных инструментов для моделирования компьютерных сетей можно выделить такие программные продукты, как Cisco Packet Tracer и GNS₃. Cisco Packet Tracer (CPT) — проприетарная программа для моделирования сетей, выпускаемая компанией Cisco Systems. GNS₃ (Graphical Network Simulator) — программный эмулятор сети с открытым исходным кодом, распространяемый под лицензией GNU GPL.

В работе [2] приведён сравнительный анализ достоинств и недостатков наиболее распространённых систем моделирования, по результатам которого наибольшим числом сильных сторон обладают

*GNU*₃ и Cisco Packet Tracer. Среди значимых преимуществ их использования выделяют дружелюбный графический интерфейс и поддержку сразу двух семейств операционных систем — Linux и Windows. Основным недостатком Cisco Packet Tracer является работа исключительно с сетевыми устройствами Cisco, в то время как в *GNU*₃ имеется возможность использовать образы виртуальных машин на различных операционных системах.

В моделировании компьютерных сетей выделяют 2 основные категории устройств: сетевые устройства, используемые для маршрутизации и передачи трафика в сетях, а также конечные устройства, которые можно использовать для генерации трафика и проверки связности сети. В качестве последних могут использоваться образы виртуальных машин, а также специальное ПО с базовыми сетевыми командами, например, проект VPCS [3].

VPCS (Virtual PC Simulator) — проект с открытым исходным кодом, который позволяет эмулировать до 9 персональных компьютеров во время моделирования сети. Небольшой объём потребляемых ресурсов по сравнению с виртуальными машинами составляет основное преимущество VPCS. Тем не менее, данного функционала вполне достаточно для задач моделирования сети. VPCS входит в комплект программного эмулятора *GNU*₃, что позволяет использовать его в проектах без дополнительной установки и настройки. Однако во время моделирования проекта компьютерной сети в *GNU*₃ была выявлена проблема невозможности установки статического маршрута по умолчанию при IPv6-адресации в виртуальном компьютере VPCS.

Для добавления функционала установки статического маршрута по умолчанию в виртуальных компьютерах VPCS требовалось доработать парсинг команды `ip` для установки IP-адреса, а также изменить логику отправки пакетов. Парсинг команды состоит из определения элементов, которые были переданы в команду: IPv6-адрес, маска и маршрут по умолчанию. При этом маска может быть не задана или записана как после IPv6-адреса, так и после маршрута по умолчанию. Также были добавлены соответствующие предупреждения, информирующие пользователя о невозможности установки введённых параметров IP-адреса (IPv6-адрес устройства не должен совпадать с маршрутом по умолчанию; маршрут по умолчанию должен находиться в той же подсети) — при данных ошибках маршрут по умолчанию игнорируется. При успешной установке IP-адреса пользователю выводится информация об установленном адресе и маршруте по умол-

чанию, если он был задан в команде, а в противном случае — установленный ранее маршрут удаляется. Различные варианты установки IP-адреса с предупреждающими сообщениями приведены в листинге: Различные варианты установки маршрута по умолчанию.

```
PC1> ip 2001:a::10/64 2001:a::1  
PC1 : 2001:a::10/64  
GATEWAY : 2001:a::1
```

```
PC1> ip 2001:a::10/64 2001:a::0::10  
Warning: gateway should not be equal to IP, gateway will be ignored  
PC1 : 2001:a::10/64
```

```
PC1> ip 2001:a::10/64 2001:b::0::1  
Warning: gateway should be in the same net, gateway will be ignored  
PC1 : 2001:a::10/64
```

При установленном статическом маршруте по умолчанию пакеты будут отправляться по данному адресу. Если MAC-адрес узла маршрута по умолчанию отсутствует в таблице MAC-адресов, то будет отправлен запрос Neighbor Solicitation для определения MAC-адреса узла. Тестирование разработанного функционала осуществлялось в программном эмуляторе *GNSS*. Для этого была спроектирована небольшая компьютерная сеть, состоящая из 2 маршрутизаторов и 2 конечных устройств, подключённых к каждому маршрутизатору соответственно (рис. 1.).

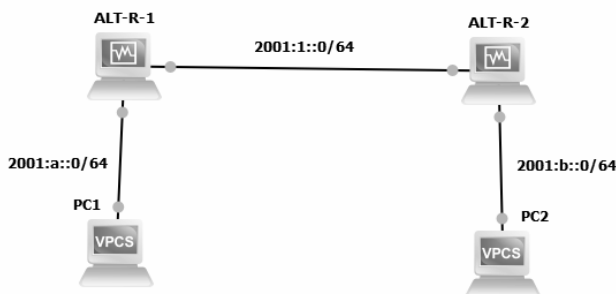


Рис. 1: Топология компьютерной сети

В качестве маршрутизаторов использовались образы виртуальных машин с установленной операционной системой «Альт Сервер»,

а в качестве конечных устройств — виртуальные компьютеры VPCS. На маршрутизаторах установлены IPv6-адреса, а также маршрут до удалённой сети. На конечных устройствах настроены IPv6-адреса и маршруты по умолчанию. Утилита ping на обоих конечных устройствах показывает, что связность присутствует (рис. 2).

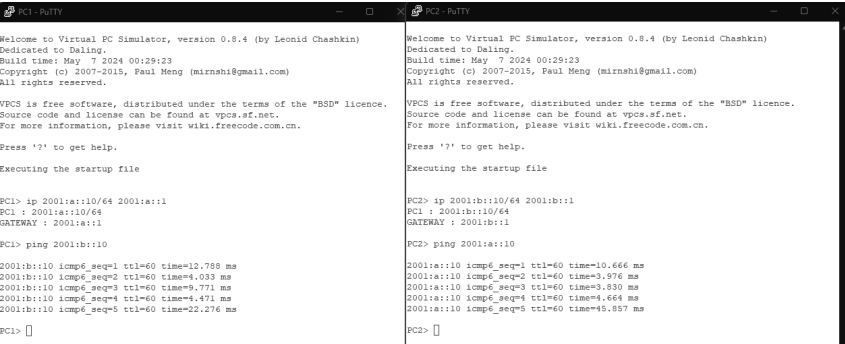


Рис. 2: Проверка связности с помощью утилиты ping.

Таким образом, можно сделать вывод, что тестирование пройдено успешно и разработанный функционал установки маршрутов по умолчанию в IPv6-адресации работает корректно на виртуальных компьютерах VPCS и может использоваться при моделировании статической IPv6-адресации.

Литература

[1] *Igulu K., Onuodu F., Singh T. P.* IPv6: Strengths and Limitations // Communication Technologies and Security Challenges in IoT: Present and Future. — Singapore: Springer Nature Singapore, 2024. — С. 147–172.

[2] *Подсадников А. В., Розов К. В., Кратов С. В.* Применение средств имитационного моделирования компьютерных сетей в учебном процессе // Информатика и образование. — 2021. — №. 1. — С. 47–56.

[3] Проект VPCS. [https://github.com/\\$GNS_3\\$/vpcs](https://github.com/GNS_3/vpcs)

Михаил Черноморец

Щёлково, Государственное бюджетное профессиональное образовательное учреждение Московской области «Щёлковский колледж», студент 2 курса

Использование инструмента *ipcalc* в рамках демонстрационного экзамена по специальности 09.02.06 «Сетевое и системное администрирование» в колледже

Аннотация

В докладе рассмотрен опыт успешного применения инструмента *ipcalc* в рамках подготовки к процедуре демонстрационного экзамена по специальности 09.02.06 «Сетевое и системное администрирование» в ГБПОУ МО «Щёлковский колледж»

Ключевые слова: *«Альт»*, *СПО*, образование, *ipcalc*, *GNU*, колледж, сети.

Общие сведения

На демонстрационном экзамене по специальности 09.02.06 «Сетевое и системное администрирование» среди прочих заданий в самом начале необходимо сделать расчёт подсетей опираясь примерно на такую схему (Рис. 1).

Конечно, мы можем для этих целей воспользоваться специальными сервисами сети Интернет, которых там в достаточном количестве. Но вот что делать, если на экзамене доступ в Интернет запрещён, а расчёты сделать необходимо. Здесь нам на помощь придёт команда *ipcalc*, которая в дистрибутиве ОС либо есть, либо её можно установить (на примере в ОС «Альт»).

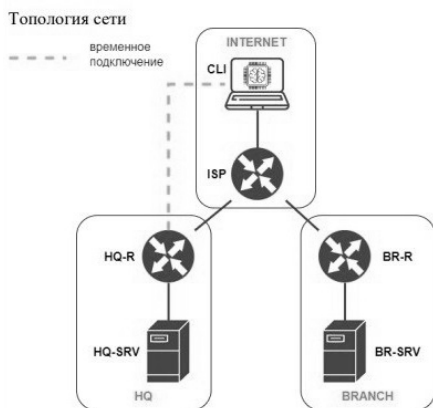
```
[user@isp ~]$ su -
```

```
Password:
```

```
[root@isp ~]# apt-get install ipcalc
```

```
Чтение списков пакетов... Завершено
```

```
Построение дерева зависимостей... Завершено
```



Задание 1 модуля 1

1. Выполните базовую настройку всех устройств:
 - а. Пул адресов для сети офиса BRANCH - не более 16
 - б. Пул адресов для сети офиса HQ - не более 64

Рис. 1: Рассчитать IP-адресацию IPv4.

Калькулятор подсетей в GNU Linux

Для того, что мы могли эффективно управлять сетью, мы должны уметь работать с подсетями. Говоря простым языком, системный администратор разбивает сети на несколько малых подсетей. Всё это является необходимым умением работы администратора, для:

- нивелирования сетевых угроз;
- улучшения состояния сетевой безопасности;
- увеличения эффективности маршрутизации.

В процедуре демонстрационного экзамена и чемпионатов «Профессионалы» это обязательный пункт, который как раз и проверяет в том или ином виде обозначенные выше компетенции. Всё это требует математических расчётов IP-адресов с масками подсетей.

Удобным инструментом для реализации данных целей является команда `ipcalc`.

Команда `ipcalc` в Linux используется для выполнения простых манипуляций с IP-адресами, таких как вычисление диапазона хостов,

широковещательной рассылки, сети и маски подстановочных знаков cisco. Он принимает сетевую маску и IPv4-адрес и возвращает полную информацию об IP-адресе хоста. Подобно другим командам Linux, она предлагает несколько параметров, определяющих, какую информацию будет содержать `ipcalc`.

Здесь мы определим, как использовать команду `ipcalc` в Linux и как с её помощью получить желаемую информацию об IP. Во-первых, давайте отобразим, какую опцию предоставляет нам команда `ipcalc` в Linux с помощью ключевого слова `--help`, а затем мы будем использовать эти опции для получения желаемого вывода. Давайте отобразим параметры команды `ipcalc` в Linux. Как обычно, мы можем вызвать справку о программе и ознакомиться с опциями программы:

```
[user@isp ~]$ ipcalc --help
```

С помощью `ipcalc` мы можем получить необходимую информацию о нужном нам сетевом адресе.

```
[user@isp ~]$ ipcalc 10.3.4.0
Address: 10.3.4.0      00001010.00000011.00000100. 00000000
Netmask: 255.255.255.0 = 24  11111111.11111111.11111111. 00000000
Wildcard: 0.0.0.255    00000000.00000000.00000000. 11111111
=>
Network: 10.3.4.0/24   00001010.00000011.00000100. 00000000
HostMin: 10.3.4.1     00001010.00000011.00000100. 00000001
HostMax: 10.3.4.254   00001010.00000011.00000100. 11111110
Broadcast: 10.3.4.255 00001010.00000011.00000100. 11111111
Hosts/Net: 254        Class A, Private Internet
```

Команда `ipcalc` возвращает IPv4-адрес сети как в десятичном, так и в двоичном формате который мы можем отсечь опцией `-b`. Для вышеуказанного адреса мы можем рассчитать необходимую нам подсеть.

```
[user@isp ~]$ ipcalc 192.168.63.1/24
Address: 192.168.63.1  11000000.10101000.00111111. 00000001
Netmask: 255.255.255.0 = 24  11111111.11111111.11111111. 00000000
Wildcard: 0.0.0.255    00000000.00000000.00000000. 11111111
=>
Network: 192.168.63.0/24 11000000.10101000.00111111. 00000000
HostMin: 192.168.63.1   11000000.10101000.00111111. 00000001
HostMax: 192.168.63.254 11000000.10101000.00111111. 11111110
Broadcast: 192.168.63.255 11000000.10101000.00111111. 11111111
Hosts/Net: 254          Class C, Private Internet
```

```
[user@isp ~]$ ipcalc 192.168.63.1/26
Address: 192.168.63.1  11000000.10101000.00111111.00 000001
Netmask: 255.255.255.192 = 26 11111111.11111111.11111111.11 000000
Wildcard: 0.0.0.63     00000000.00000000.00000000.00 111111
=>
Network: 192.168.63.0/26 11000000.10101000.00111111.00 000000
HostMin: 192.168.63.1   11000000.10101000.00111111.00 000001
HostMax: 192.168.63.62  11000000.10101000.00111111.00 111110
Broadcast: 192.168.63.63 11000000.10101000.00111111.00 111111
Hosts/Net: 62           Class C, Private Internet
```

Давайте познакомимся, с параметрами команды.

- -s — используется для получения необходимого количества хостов в одной подсети. Например, 10.

```
[black@tpad compare]$ ipcalc 192.168.63.1 -s 10
Address: 192.168.63.1      11000000.10101000.00111111. 00000001
Netmask: 255.255.255.0 = 24 11111111.11111111.11111111. 00000000
Wildcard: 0.0.0.255      00000000.00000000.00000000. 11111111
=>
Network: 192.168.63.0/24  11000000.10101000.00111111. 00000000
HostMin: 192.168.63.1    11000000.10101000.00111111. 00000001
HostMax: 192.168.63.254  11000000.10101000.00111111. 11111110
Broadcast: 192.168.63.255 11000000.10101000.00111111. 11111111
Hosts/Net: 254           Class C, Private Internet

1. Requested size: 10 hosts
Netmask: 255.255.255.240 = 28 11111111.11111111.11111111.1111 0000
Network: 192.168.63.0/28  11000000.10101000.00111111.0000 0000
HostMin: 192.168.63.1    11000000.10101000.00111111.0000 0001
HostMax: 192.168.63.14   11000000.10101000.00111111.0000 1110
Broadcast: 192.168.63.15 11000000.10101000.00111111.0000 1111
Hosts/Net: 14           Class C, Private Internet

Needed size: 16 addresses.
Used network: 192.168.63.0/28
Unused:
192.168.63.16/28
192.168.63.32/27
192.168.63.64/26
192.168.63.128/25
```

Команда `ipcalc` вычислила подсеть с 10 хостами, с масками, которые нужно указывать при работе с заданием.

Выводы

По результатам апробации команды `ipcalc` при подготовке к демонстрационному экзамену по специальности 09.02.06 «Сетевое и системное администрирование» сделан вывод, что данный вид заданий довольно успешно реализуется в среде свободного программного обеспечения с помощью программы консольной `ipcalc`, которая имеется в репозиториях практически любого дистрибутива Linux.

Андрей Вахранев

Волгоград, Волгоградский филиал РГУ СоцТех

Проект: Внедрение в учебный процесс Lubuntu, Calculate, Alt

Свободное программное обеспечение в Волгоградском филиале РГУ СоцТех или «История одной аудитории»

Аннотация

Внедрение и использование СПО в Волгоградском филиале РГУ
СоцТех

Ключевые слова: *Lubuntu, Calculate, Alt.*

История внедрения СПО в данном образовательном учреждении среднего профессионального образования насчитывает более 10 лет и пережила с самой организацией не одну реорганизацию.

До 2012 года был Волгоградский государственный колледж профессиональных технологий, экономики и права. Примерно с 2009 г., благодаря определённым обстоятельствам, в одной из аудиторий пришлось внедрять СПО. Тогда это был дистрибутив Lubuntu 32-разрядный для студенческих компьютеров (Pentium 3/4, 512/1ГБ ОЗУ, 80ГБ HDD) и Ubuntu Server 32-разрядный для файлов и обновлений (Athlon 1GHz, 258MB ОЗУ). Преподавались разнообразные дисциплины, такие как «Операционные системы», «Компьютерные сети», «Архитектура ЭВМ» и т.п. с использованием таких средств, как VirtualBOX, LibreOffice и т.п.

После того как ушло федеральное финансирование среднего профессионального образования, ВГКПТЭиП присоединился к Московскому государственному гуманитарно-экономическому институту и стал его филиалом. С тех пор регулярно стали меняться директора и вся остальная администрация, аудиторию под управлением Linux-based систем регулярно старались перевести на Windows XP. С переменным успехом. В том числе потому, что ремонтом (апгрейдом) компьютерного парка приходилось заниматься самостоятельно. Иногда ПК просто переставали справляться со своими задачами.

Затем МГГЭи реорганизовался в МГГЭУ (стал университетом), а в Волгоградском филиале продолжали работать по программам среднего профессионального образования. Одни специальности закрывались, другие открывались. Затем пришло письмо МинОбрНауки РФ

от 10.08.2017 №03-1463 по переходу федеральных органов исполнительной власти и государственных, внебюджетных фондов на использование отечественного ПО.

Из списка Росреестра была выбрана единственная абсолютно бесплатная, свободная и открытая система — Calculate Linux. К тому времени некоторые компьютеры уже были оснащены 64-разрядными ЦПУ, до 2ГБ ОЗУ. В декабре 2018 Calculate перестал поддерживать 32-разрядные платформы. Где-то с 2021 года от «Базальт СПО» учебному заведению начали поступать предложения использовать ОС «Альт» в рамках программы «Альт Академия», но убедить директора в целесообразности воспользоваться предложением не вышло (аргумент: нигде же эти Linux не используются). В сентябре 2022 г. по приказу нового директора в аудитории изъяли на списание мониторы (ЛТ). Весь учебный год прошёл с использованием «вражеских» систем в других аудиториях.

В течение 2023 года в рамках личной переписки с edu@basealt.ru проходили попытки направить предложения новому директору с тем, чтобы убедить её исполнить приказ правительства по импортозамещению. Ведь какие-то приказы исполняются, и прямо с каким-то рвением. А тут — массовый саботаж и никто ничего не замечает. В итоге, к концу 2023 от местного «начальника IT-отдела» была получена отмашка ставить «Альт», благо в какую-то аудиторию поступили новые компьютеры с новыми мониторами, на которых, естественно, развернули импортные системы, зато освободилось 13 мониторов для использования в аудитории с СПО. Благодаря Ali Express и Авито были дособраны 64-разрядные станции с 4ГБ ОЗУ под «Альт Образование».

К маю 2024 года оказалось, что договор с «Альт Академией» так и не был заключён. Но, благодаря переписке (огромная благодарность Быховскому Ярославу), получено официальное разрешение использовать «Альт Образование» и «Альт Сервер» на период, которого должно хватить для официального оформления полноценной программы «Альт Академии».

Также можно поделиться тематическим планированием междисциплинарного курса «Операционные системы», в рамках которого имеются темы, построенные на «Альте»:

1. Подготовка Linux-based дистрибутива. Установка.

2. Управление учётными записями пользователей и доступом к ресурсам.
3. Управление приложениями в Linux-based системах.
4. Компиляторы и интерпретаторы.
5. Серверная Linux-based система.
6. Работа с сетевой файловой системой.
7. Аудит событий системы.
8. Изучение средств защиты информации в операционных системах.

На youtube-канале Fentor¹ имеется плейлист² с демонстрацией основных моментов выполнения работ.

Дмитрий Костюк, Анастасия Маркина, Антип Шульган
Брест, Брестский государственный технический университет
<https://bitbucket.org/AsyaAliset/materialy-po-python/downloads/>

Адаптация университетской программы по алгоритмизации и программированию на языке Python для дополнительного довузовского образования

Аннотация

Приводится опыт модификации базового курса алгоритмизации и программирования, рассчитанного на первую ступень высшего образования специальностей профиля информатики и радиоэлектроники, для школьников.

Рассмотрены особенности представления основных конструкций, абстракций данных, а также базовых алгоритмов на языке Python, использованные для нивелирования низких показателей саморегуляции и концентрации, дробление тем на наборы малых временных интервалов для адаптации к способностям удержания фокуса внимания обучаемых. Рассмотрена специфика применения средств разработки, распространяемых под свободными лицензиями, в рамках курса.

¹<https://www.youtube.com/@Fentor>

²<https://www.youtube.com/playlist?list=PLbz4ZPZ0b1FSvQCmoCXTrjyzjfTbxUwnP>

Ключевые слова: *методика преподавания, Python, довузовская подготовка.*

В последние годы наблюдается тенденция снижения возраста вхождения в программирование. В отличие от прежнего подхода с обучением 12-летних детей блочному программированию в Scratch, возраст обучаемых начинается с 10 лет и менее, что предъявляет к учебным материалам для подростков новые требования. Ниже рассматриваются опыт адаптации учебной программы «Алгоритмизация и программирование на языке Python» Брестского Государственного Технического Университета для подростков в рамках финального этапа изучения связки из трёх языков программирования: Scratch, Lua, и Python.

При адаптации курса приходилось учитывать, что скорость письма, чтения и объёма воспринимаемого учебного материала у детей 13–16 лет существенно ниже, чем у студентов, даже без учёта скорости набора текста. Когнитивные и эмоциональные особенности обучаемых, связанные с тем, что у них ещё не полностью сформирована префронтальная кора, ответственная за планирование, принятие решений и социальное поведение, требуют специальных педагогических подходов [1]. Помимо адаптации для детей, преподавание данного курса потребовало адаптации для их родителей, которые хотят, чтобы их дети получали качественные знания, и при этом в своей основной массе либо имеют отношение к сфере ИТ сами, либо имеют программистов в близком окружении. Это накладывает дополнительные требования к прозрачности и однозначности как самой учебной программы, так и форм промежуточного контроля.

Для подростков и студентов особенно важно развитие метакогнитивных навыков — способности осознавать и контролировать свои собственные когнитивные процессы, включая саморегуляцию (способность контролировать свои эмоции, мысли и действия), постановку целей (умение ставить конкретные, достижимые цели и планировать шаги для их достижения) и рефлексии (способность анализировать свой опыт, действия и результаты, чтобы извлекать уроки и улучшать свою деятельность в будущем). Это помогает им справляться с возрастающими академическими и социальными требованиями [2].

Ключевую роль в обучении для целевой аудитории играет эмоциональная регуляция, а развитие эмоционального интеллекта способствует лучшему усвоению материала [3][4]. Создание учебной среды, учитывающей эмоциональные потребности учеников, помогает улучшить их результаты и общее благополучие в случае, когда процесс

преподавания учитывает нормотипичные этапы развития префронтальной коры обучаемых:

1. Средний школьный возраст (6–12 лет). Синаптический прунинг заключается в «обрезке» лишних нейронных связей, что помогает сделать мозг более эффективным. Этот процесс усиливается в возрасте от 6 до 12 лет, дети начинают развивать навыки планирования, организации, самоконтроля и решения проблем.
2. Подростковый возраст (12–18 лет). Продолжается процесс значительных изменений в префронтальной коре. Период характеризуется высокой нейропластичностью, что означает возможность значительных изменений и адаптаций в нейронных связях. В это время префронтальная кора развивается медленнее по сравнению с миндалевидным телом, которое отвечает за эмоции. Такое несоответствие приводит к тому, что эмоциональная реактивность подростков часто превышает их способность к рациональному мышлению и самоконтролю [1][4].
3. Молодость и взрослая жизнь (18–25 лет). Префронтальная кора продолжает развиваться до 24–25 лет. Один из ключевых процессов — миелинизация — улучшает скорость и эффективность передачи нервных импульсов. По мере завершения развития префронтальной коры усиливаются и стабилизируются нейронные пути, что способствует улучшению когнитивных функций, таких как планирование, принятие решений, социальное поведение и контроль импульсов [3][4].

С учётом возрастных особенностей обучаемых, при составлении планов уроков и программы курса программирования в целом приходится учитывать высокую нейропластичность детей, среднее развитие миндалевидного тела, необходимость создания поддерживающей учебной среды, стратегии успеха, развития метакогнитивных навыков, поддержки интерактивной учебной среды и индивидуального подхода. Существенную роль играет также нормативно-правовую база и требования к проведению уроков естественно-математической направленности [6][7][8].

Курс «Основы алгоритмизации и программирования на языке Python» включает два годичных этапа: на первом году преподаются непосредственно особенности языка и алгоритмизация, на втором году обучения — основы ООП. Каждый год рассчитан на 72 учебных

часа, включающих в себя как теоретическую, так и практическую часть.

Тематически в курсе можно выделить три части. В первой делается акцент делается на построение алгоритмов, составление программ на их основе, изучение базовых основ языка программирования. Раздел нацелен на развитие алгоритмического мышления и усвоения базисных парадигм программирования с их апробацией на поставленных практических задачах. Развивает самостоятельность, логическое мышление и сообразительность, творческое отношение к процессу обучения, умение анализировать и применять полученные теоретические знания на практике.

Во второй части акцент переходит на средства визуального программирования приложений с использованием основных принципов GUI и изучение работы разнообразных виджетов из двух пакетов, ориентированных на работу с графикой. Данная часть ориентирована на развитие у обучающихся потребности в творческой деятельности, стремление к самовыражению через компьютерные технологии и программирование.

Третья часть смещает акцент на самостоятельную и коллективную работу обучающихся над творческими проектами, что воспитывает навыки самоорганизации и самоконтроля, работы в коллективе.

Практические задания построены на быстрой визуальной демонстрации результата (необходим набор кода до 50–70 строк), разных уровней сложности, что реализует индивидуальный подход к учащимся и стратегию успеха (базовые задачи решает каждый ребёнок, самостоятельно или с помощью преподавателя). В дополнение к этому используется принцип командной работы, когда обучающиеся выполняют групповое задание в малых группах (до 3 человек, проект до 3 занятий). Промежуточный контроль выполняется в форме квизов с элементами геймификации, когда преподаватель может давать бонусы, а игроки могут использовать различные усиления.

Помимо собственно интерпретатора Python, используемые инструментальные средства включают библиотеку Qt, с помощью которой обучаемыми создаётся графический интерфейс, сред разработки QtCreator и PyCharm Community Edition. Также применяются графические пакеты turtle и Tkinter. Учебно методический комплекс курса находится в открытом доступе и содержит учебную программу курса, методические рекомендации преподавателям с теоретическим и практическим материалом, средствами и способами про-

межуточного контроля, рабочую тетрадь для обучающихся с тезисным теоретическим материалом.

Литература

- [1] Развитие ребенка / Психология ребенка [Электронный ресурс] — <https://childdevelop.info/articles/psychology/440/> Дата обращения: 08.06.2024;
- [2] Психологическая газета / Метакогнитивная компетенция. <https://psy.su/feed/3966/> Дата обращения: 08.06.2024;
- [3] Солнечный свет / Этапы и особенности развития мозга ребенка URL: <https://solncesvet.ru/blog/baza-znaniy/razvitiye-mozga-rebenka/> Дата обращения: 09.06.2024.
- [4] Мозгмед / Развитие мозга ребенка и его особенности URL: <https://mozgmed.ru/struktura/razvitie-mozga-rebenka> Дата обращения: 09.06.2024.
- [5] МедУнивер / Предфронтальная кора мозга: функции, признаки поражения лобной доли URL: https://meduniver.com/Medical/Neurology/predfrontalnaia_kora_mozga.html Дата обращения: 10.06.2024.
- [6] Кодекс Республики Беларусь об образовании. — Минск: Нац. центр правовой информ. Респ. Беларусь, 2011. — 400с. // Национальный реестр правовых актов РБ №2/1795 от 17.01.2011 // Национальный правовой интернет-портал Республики Беларусь — 2011.
URL: http://www.pravo.by/world_of_law/text.asp?RN=hk1100243/ — Дата доступа: 03.06.2024.
- [7] Национальная стратегия устойчивого социально-экономического развития Республики Беларусь на период до 2030.
URL: <http://www.economy.gov.by/uploads/files/NSUR2030/Natsionalnaja-strategijaustojchivogo>. — Дата доступа: 03.06.2024.
- [8] Сборник. Программы объединений по интересам дополнительного образования детей и молодежи эколого-биологического профиля: для руководителей объединений по интересам учреждений общего среднего образования и дополнительного образования детей и молодежи эколого-биологического профиля. Повышенный уровень / Л. П. Калиновская и др.; — Минск: ПДУП «Типография Федерации профсоюзов Беларуси», 2015. — 69 с.

Роман Крашенинников, Иван Митричев

Москва, ФГБОУ ВО «Российский химико-технологический университет имени Д. И. Менделеева»

Проект: Библиотека для решения фармакокинетических задач PyPharm
<https://pypi.org/project/pypharm/>

Разработка библиотеки Python для решения задач фармакокинетики

Аннотация

В последние годы наблюдается активное увеличение использования моделирования при разработке лекарств. Особую роль играет исследование фармакокинетики разрабатываемых препаратов. Подбор параметров фармакокинетических моделей является задачей многомерной оптимизации, с которой эффективно справляются эволюционные алгоритмы. В данной работе представлена новая открытая библиотека на языке Python, позволяющая производить моделирование и подбор параметров фармакокинетических моделей с применением эволюционных алгоритмов.

Ключевые слова: *эволюционные алгоритмы, генетический алгоритм, алгоритмы, вдохновлённые природой, фармакокинетика.*

Фармакокинетика

С каждым годом моделирование всё чаще применяется в разработке лекарственных препаратов, что подтверждается ростом числа упоминаний о нём в различных источниках. Эффективность такого подхода, в частности, подтверждается экономической выгодой [1, 2]. Одна из областей применения моделирования — разработка фармакокинетических моделей. Фармакокинетика (ФК) — это наука, изучающая скорость движения лекарственного средства и/или его метаболитов в организме и силы, действующие на этот процесс.

Фармакокинетические модели помогают подобрать расчётные параметры для интерполяции или экстраполяции результатов, что способствует подбору корректного режима дозирования препарата [3]. Распределение препарата во внесосудистые области, такие как внутриклеточная жидкость или ткани, является быстрым и обратимым процессом. Выведение препарата происходит путём экскреции или биотрансформации в тканях или плазме.

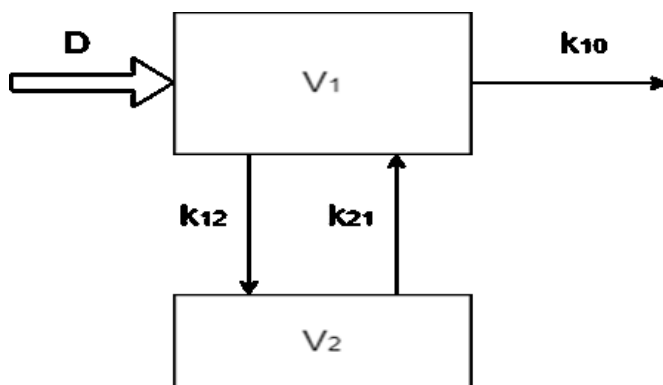


Рис. 1: Двухкамерная модель [5].

Для описания процесса высвобождения препарата используются камерные модели. Они представляют организм как набор отдельных камер, содержащих часть введённой дозы препарата, а скорость перемещения препарата из одной камеры в другую определяется константами скорости [4].

Принцип камерных моделей может быть продемонстрирован на наиболее распространённом их виде — двухкамерной модели (рис. 1). Она может быть описана математически в виде системы дифференциальных уравнений [6]. Камерные модели отличаются простотой, поэтому исследователи часто создают своё внутреннее программное обеспечение для моделирования ФК. Однако существуют и общедоступные программы, например, *GastroPlus* и *Simcyp PBPK Simulator*, которые хорошо зарекомендовали себя в этой области. Они способны строить различные фармакокинетические модели с высокой точностью [6, 7]. *GastroPlus* и *Simcyp PBPK Simulator* являются дорогостоящими и зарубежными, что ограничивает возможности исследователей на территории России. В связи с этим было принято решение создать продукт с открытым исходным кодом, который позволит создавать и оптимизировать фармакокинетические модели. В основу ПО предполагается заложить общедоступную библиотеку на языке *Python*. Для поиска параметров моделей предполагается использовать эволюционные алгоритмы оптимизации.

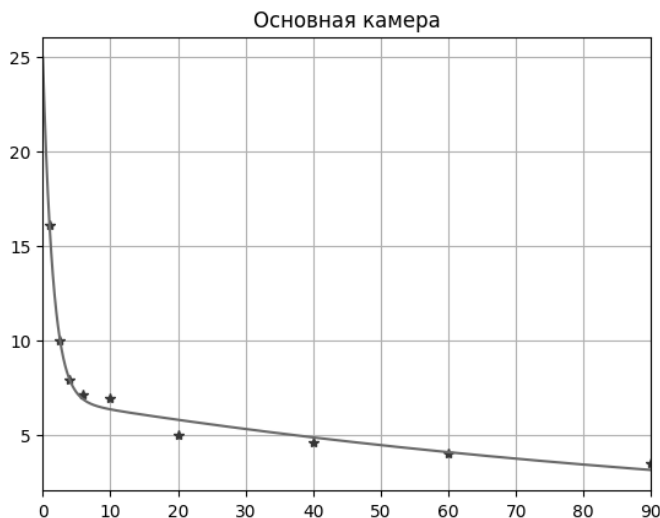


Рис. 2: Визуализация построения модели по экспериментальным данным.

Библиотека PyPharm

Созданная в ходе работы библиотека **PyPharm** способна моделировать фармакокинетику с использованием камерных моделей, а также производить поиск настроечных параметров по экспериментальным данным. Основным модулем библиотеки является модуль **models**, который на текущий момент имеет в себе три типа моделей, отражённых в соответствующих классах:

- **BaseCompartmentModel** — классическая камерная модель;
- **MagicCompartmentModel** — классическая камерная модель с дополнительно введённым параметром **magic_coefficient**, отвечающим за преобразование единиц измерения входной концентрации в единицы измерения выходной концентрации. Этот параметр необходим, в тех случаях, когда доза препарата задана в одних единицах измерения, а концентрация в органах — в других, и нет явного способа преобразования между ними;

- `ReleaseCompartmentModel` — камерная модель, учитывающая высвобождение лекарственного вещества из носителя по заданному закону.

Два последних класса наследуются от `BaseCompartmentModel`, поэтому основные методы классов подобны и отличаются лишь набором входных параметров. Далее будет описана суть их работы. Библиотека поддерживает построение ФК-моделей заданной конфигурации, а также оптимизацию параметров модели с использованием эволюционных алгоритмов, таких как алгоритм взаимодействующих стран [4, 7, 8] и генетический алгоритм (рис. 2).

Таким образом, в рамках данного исследования разработана библиотека `PyPharm`, способная решать задачи построения фармакокинетических моделей. На данный момент библиотека имеет версию 1.3.6 и доступна по ссылке <https://pypi.org/project/pypharm/>. Там же имеются примеры использования библиотеки и описание её возможностей. Библиотека способна строить три вида камерных моделей и уже была опробована и хорошо показала себя в различных исследованиях [4, 8]. В дальнейшем планируется расширение `PyPharm`, с добавлением в неё методов машинного обучения и расширением списка доступных моделей.

Литература

- [1] EMA. Guideline on the qualification and reporting of physiologically based pharmacokinetic (PBPK) modelling and simulation. https://www.ema.europa.eu/en/documents/scientific-guideline/guideline-reporting-physiologically-based-pharmacokinetic-pbpk-modelling-and-simulation_en.pdf
- [2] Hill-McManus D, Marshall S, Soto E, Lane S, Hughes D. Impact of non-adherence and flare resolution on the cost-effectiveness of treatments for gout: application of a linked pharmacometric pharmacoeconomic model. *Value Health*, 2018; 21(12): 1373–1381.
- [3] Yoon E, Babar A, Choudhary M, Kutner M, Prysopoulos N. Acetaminophen-induced hepatotoxicity: a comprehensive update. *J Clin Transl Hepatol*, 2016; 4(2) 131–142.
- [4] Крашенинников Р. С., Митричев И. И. Применение алгоритма взаимодействующих стран для поиска констант фармакокинетической модели // Математические методы в технологиях и технике. — 2023. — № 4. — С. 12–15. DOI 10.52348/2712-8873 MMTT 2023.02.12

- [5] Официальный сайт GastroPlus. <https://www.simulations-plus.com/software/gastroplus/>
- [6] Официальный сайт Simcyp PBPК. <https://www.certara.com/software/simcyp-pbpk/>
- [7] Крашенинников Р. С., Митричев И. И. Оптимизационный алгоритм взаимодействующих стран // Информационные технологии в моделировании и управлении: подходы, методы, решения: Материалы VI Всероссийской научной конференции с международным участием, Тольятти, 18–20 апреля 2023 года. — Тольятти: Тольяттинский государственный университет, 2023. — С. 184–191.
- [8] Крашенинников Р. С., Митричев И. И. Оптимизационный алгоритм взаимодействующих стран и его применение для решения задач фармакокинетики // Прикладная математика и вопросы управления. — 2023. — № 4. — С. 126–135. — DOI 10.15593/2499-9873/2023.4.08

Сергей Мартишин, Марина Храпченко

Москва, Институт системного программирования им. В. П. Иванникова РАН

Представление конфиденциальных данных в агрегированном виде с использованием векторного формата SVG, языков R и Rust

Аннотация

Рассмотрены вопросы визуализации конфиденциальных данных в агрегированном виде для выполнения дедуктивно безопасных запросов. Обоснован выбор языка R для представления статистических данных в формате SVG и языка Rust для создания клиентской части и запросов к базе данных.

Ключевые слова: СПО, дедуктивная безопасность данных, визуализация данных, язык Rust, язык R.

В научных проектах, имеющих практическое применение и использующих большие массивы данных, чаще всего хранение и обработка данных происходит с использованием облачных сервисов. Это позволяет обеспечить совместное использование данных и сэкономить на высокопроизводительном оборудовании для вычислений.

Однако в некоторых случаях возникает необходимость работы с конфиденциальными данными, например, медицинскими. Это обуславливает определённые требования к приложениям, осуществляющих обработку такого рода данных, основным из которых является невозможность раскрытия конфиденциальной информации на любом этапе взаимодействия с облаком.

Ранее авторами в [1] была разработана и исследована модель облачных вычислений, включающая традиционные компоненты: облачный сервер, пользователей (хранящих свои данные на облаке), клиентов (производящих вычисления с этими данными), криптосерверы и сети связи. Помимо традиционных компонентов модель также включает разрешённые вычисляемые функции и центр аутентификации. Задачей центра аутентификации является проверка полномочий клиента.

В [2] было сформулировано определение дедуктивной безопасности, подразумевающее, что клиент базы данных, получив ответы на все доступные ему запросы, не может получить дедуктивно (в виде логического следствия) конфиденциальные данные пользователей. Было доказано, что симметрические запросы являются дедуктивно безопасными, поскольку призваны собирать лишь статистические данные. Если клиент не обладает специальными полномочиями, то разрешения ему выдаются на основе дедуктивной безопасности. В этом случае, данные, получаемые клиентом, имеют агрегированный вид и не содержат персональных данных.

Одной из главных задач приложения, обрабатывающего агрегированные данные, является получение результатов вычислений в виде, удобном для восприятия человеком. Визуализация агрегированных данных позволяет оценить общую картину, выявить закономерности и детально проанализировать поведение исследуемых величин.

В статьях [3, 4] авторы рассматривали вопрос работы с агрегированными конфиденциальными данными и их визуализации. В [3] был обоснован выбор интерпретируемого объектно-ориентированного языка R [5] как одного из наиболее удобных используемых языков программирования для анализа и визуализации статистических данных. Язык R является СПО (свободное программное обеспечение) с открытым исходным кодом в рамках проекта GNU.

Для того чтобы приложение работало без сбоев, проблем с производительностью и простоев, используется менеджер процессов `pm2` [6]. Он имеет встроенный балансировщик нагрузки, позволяет

следить за потребляемыми ресурсами запущенных процессов, автоматически перезапускать процессы после системного сбоя и т. д. Распространяется под лицензией AGPL-3.0.

Существует два принципиально различных подхода к представлению графической информации: растровая и векторная графика. В случае использования растровой графики (изображение представлено в виде пикселей) визуализация больших данных представляет сложность при отображении тех или иных участков изображения: мелкий текст может становиться нечитаемым, изображения могут накладываться друг на друга и т. д. Векторная графика основана на математическом описании элементарных геометрических объектов. Если информация об объекте хранится в векторной форме, то изображение при выводе на устройство отображения можно масштабировать и изменять (вращать, перемещать и т. п.) и качество изображения при масштабировании не ухудшается. Именно поэтому для на практике всё чаще всего используется формат **SVG**.

Для создания серверной и клиентской частей и запросов к базе данных используется язык **Rust** [7]. **Rust** является свободным ПО. Язык ориентирован на использование многоядерных процессоров и позволяет запускать независимые процессы одновременно.

Основными достоинствами **Rust** являются:

- статическое управление памятью во время компиляции (отсутствие неинициализированных переменных, случайных утечек памяти, гонки данных между потоками и др.);
- отсутствует неопределённое поведение во время выполнения;
- наличие полезных языковых особенностей, таких как перечисления и сопоставление с образцом;
- развитый инструментарий, включающий развёрнутые сообщения компилятора, содержащие информацию об ошибках и способах их устранения, встроенный менеджер зависимостей, встроенная поддержка тестирования и пр.

Для серверной части предлагается использование фреймворка **Actix Web** [7]. Он позволяет встраивать в приложения на языке **Rust** функциональность **http**-сервера и клиента. К достоинствам **Actix Web** относятся его производительность, безопасность, включая безопасность памяти и параллелизма, поддержка асинхронных запросов.

Actix Web предоставляет различные примитивы для создания веб-серверов и приложений с помощью Rust, обеспечивает маршрутизацию и обмен информацией между клиентом и сервером.

Использование языка R для визуализации агрегированных конфиденциальных данных в формате SVG позволяет детально анализировать графическую информацию. Векторный формат уменьшает объём данных, передаваемых по сети интернет. Наличие в языке R большого количества встроенных статистических функций делает его использование удобным в научных проектах, связанных с обработкой конфиденциальных данных и их визуализации. Язык Rust позволяет реализовать серверную часть приложения, осуществлять запросы к базам данных и к серверу R.

Код проекта доступен по адресу: https://github.com/otd13isp/Pereslav12024_1

Литература

- [1] Варновский Н. П., Мартишин С. А., Храпченко М. В., Шокуров А. В. Пороговые системы гомоморфного шифрования и защита информации в облачных вычислениях, Программирование. 2015. № 4. С. 47–51.
- [2] Варновский Н. П., Захаров В. А., Шокуров А. В. О дедуктивной безопасности запросов к базам конфиденциальных данных в системе облачных вычислений // Вестник Московского университета. Серия 15: Вычислительная математика и кибернетика. 2017. № 1. С. 38а–44.
- [3] Мартишин С. А., Храпченко М. В. Визуализация агрегированных конфиденциальных данных в научных проектах в сети Интернет с использованием языка R. // Объединённая конференция «СПО: от обучения до разработки»: материалы конференции / Переславль-Залесский, 19–22 мая 2022 г. / отв. ред. Чёрный В. Л. — М.: МАКС Пресс, 2022. — 208 с. ISBN 978-5-317-06790-8. с. 201–205.
- [4] Мартишин С. А., Храпченко М. В. Реализация агрегированного графического запроса конфиденциальных данных из облачной базы данных с использованием языков Go и R. // Восемнадцатая конференция «Свободное программное обеспечение в высшей школе» Переславль-Залесский, 27–29 января 2023 г. Москва, МАКС Пресс, 2023, ISBN 978-5-317-06921-6 <https://doi.org/10.29003/m3136.978-5-317-06921-6>. с. 128–132.

- [5] The R Project for Statistical Computing [Электронный ресурс] — Режим доступа: <https://www.r-project.org>, свободный. Яз. англ. — Дата обращения: 13.05.2023.
- [6] PM2 [Электронный ресурс] — Режим доступа: <https://pm2.io>, свободный. Яз. англ. — Дата обращения: 13.05.2023.
- [7] Rust [Электронный ресурс] — Режим доступа: <https://www.rust-lang.org>, свободный. Яз. англ. — Дата обращения: 13.05.2023.
- [8] Actix [Электронный ресурс] — Режим доступа: <https://actix.rs>, свободный. Яз. англ. — Дата обращения: 13.05.2023.

Сергей Мартишин, Марина Храпченко

Москва, Институт системного программирования им. В. П. Иванникова РАН

Распараллеливание поисковых запросов к веб-серверу с использованием языка Rust

Аннотация

Рассматриваются программные средства для создания параллельных запросов для поиска информации с использованием регулярных выражений. Описаны принципы подготовки текстовых файлов для осуществления поиска в них. Приведена ссылка на код проекта, на основе которого можно создать проект, позволяющий осуществлять параллельный поиск в базах данных.

Ключевые слова: СПО, дедуктивная безопасность данных, визуализация данных, язык *Rust*, язык *R*.

Создание приложений, которые имеют практическое применение, является одним из важных навыков, которыми должны овладеть студенты в процессе обучения информационным технологиям. В процессе разработки приложений студентам необходимо выбрать программное обеспечение, позволяющее реализовать все требования к приложению наилучшим образом. Чаще всего естественным выбором является СПО.

СПО — свободное программное обеспечение, которое широко используется не только потому, что является бесплатным. Открытость кода и участие в его создании, тестировании, применении большого

количества разработчиков и пользователей приводит к тому, что качество кода СПО находится на очень высоком уровне в плане функциональности, надёжности, возможности использования на различных платформах.

Также необходимо заметить, что современное аппаратное обеспечение в основном базируется на многоядерных процессорах, поэтому одновременное (параллельное) выполнение нескольких задач является естественным решением увеличения скорости обработки информации. Надёжность работы многопоточных программ также зависит от организации безопасной работы с памятью. Соответственно, в процессе обучения студентам необходимо давать знания о современных средствах СПО: языках программирования, которые имеют встроенные средства распараллеливания и механизм управления памятью. Язык программирования Rust [1] удовлетворяет этим требованиям.

Основными достоинствами языка являются:

- статическое управление памятью во время компиляции, то есть отсутствие неинициализированных переменных и случайных утечек памяти;
- отсутствие NULL-указателей и гонки данных между потоками;
- отсутствие неопределённого поведения во время выполнения (например, доступ к массиву проверяется по границам);
- наличие языковых особенностей таких, как перечисления и сопоставление с образцом;
- развитый инструментарий, включающий развёрнутые сообщения компилятора, содержащие информацию об ошибках и способах их устранения, встроенный менеджер зависимостей, встроенная поддержка тестирования.

С помощью языка Rust студенты могут создавать многопоточные приложения, например, проект параллельного поиска информации.

В качестве операционной системы (ОС) была выбрана одна из самых простых в установке и использовании ОС Linux Fedora с открытым исходным кодом в рамках проекта GNU. Для поиска информации использовалась стандартная процедура лемматизации текста. Лемма (англ. lemma) — начальная, словарная форма слова. Например, в русском языке словарной формой считается: существительные — именительный падеж, единственное число. Лемматизация текста заключается в приведении каждого слова в нормальную морфологическую форму.

Для учебного проекта студентам предлагается для лемматизации использовать широко распространённый пакет `rumystem3` языка Python [2] (СПО). Если Python не установлен, необходимо сначала его установить. Загрузить пакет можно командой: `pip install rumystem3`.

Поиск будет производиться в текстовых файлах. В процессе предобработки представим каждую строку файла в виде двух частей. Первая часть строки будет предназначена для поиска, а вторая часть — для вывода на экран. Первая часть строки будет содержать исходный текст после предварительно выполненной лемматизации, сортировки лемм в алфавитном порядке и перевода букв в нижний регистр. Вторая часть строки содержит краткое описание сайта на языке HTML для вывода на экран.

Поиск осуществляется с использованием регулярных выражений. Упаковка `regex` из библиотеки `crates.io` предоставляет процедуры для поиска совпадений в строках текстовых файлов при помощи регулярных выражений.

В качестве веб-сервера студентам предлагается использовать веб-фреймворк `Actix Web` [3] для Rust. Помимо того, что он является одним из наиболее быстрых фреймворков, он поддерживает асинхронные обработчики запросов, а также имеет большой набор библиотек для обработки форм, обработки файлов, тестирования и пр.

Для обмена данными между клиентом и сервером используется `XMLHttpRequest` — API (application programming interface). Он позволяет веб-браузерам асинхронно взаимодействовать с веб-сервером, то есть делать HTTP-запросы к серверу без перезагрузки страницы [4].

Поскольку проект является учебным, то для простоты реализации студентам предложено подготавливать строку запроса при помощи библиотеки `rumystem3` языка Python. Строка для поиска будет лемматизирована и отсортирована. Приведение букв в строке поиска к нижнему регистру осуществляется в коде основного файла проекта на Rust.

Для поиска совпадений в различных текстовых файлах в коде на Rust создаются независимые потоки (для поиска в каждом файле в своём потоке), и поиск выполняется параллельно. В рамках расширения проекта студенты могут осуществить поиск по базам данных (MySQL, DuckDB и пр.), используя аналогичный порядок действий. Результат поиска можно поместить в базу «ключ–значение» для быстрой последующей выдачи результатов.

Работа над проектом позволяет студентам освоить написание параллельных поисковых запросов к веб-серверу, изучить основы языка Rust, получить опыт работы с фреймворком Actix Web. Эти навыки будут полезны на практике и помогут студентам стать востребованными специалистами.

Код проекта доступен по адресу: https://github.com/otd13isp/Pereslav12024_2.

Литература

- [1] Rust [Электронный ресурс] — Режим доступа: <https://www.rust-lang.org>, свободный. Яз. англ. — Дата обращения: 27.05.2023.
- [2] Python [Электронный ресурс] — Режим доступа: <https://www.python.org>, свободный. Яз. англ. — Дата обращения: 27.05.2023.
- [3] Actix [Электронный ресурс] — Режим доступа: <https://actix.rs>, свободный. Яз. англ. — Дата обращения: 27.05.2023.
- [4] Standart ECMA-262 14th Edition/June 2023 [Электронный ресурс] — Режим доступа: <https://www.ecma-international.org/publications-and-standards/standards/ecma-262>, свободный. Яз. англ. — Дата обращения: 27.05.2023.

Григорий Гриценко, Виталий Фраленко

Переславль-Залесский, Аспирантура ИПС им. А. К. Айламазяна РАН

Использование открытого программного обеспечения в задачах обнаружения объектов в видеопотоке

Аннотация

Целью настоящей работы является исследование открытого программного обеспечения, позволяющего решать задачи обнаружения целевых объектов. Приводятся результаты проведения обзора проектов открытого программного обеспечения для решения задач обнаружения и трекинга объектов с использованием нейронных сетей с анализом эффективности качества работы разработчиков.

Ключевые слова: *свободное программное обеспечение, обнаружение объектов.*

Задача распознавания образов является одной из самых популярных и перспективных, для её решения разрабатываются различные математические аппараты с соответствующими программными реализациями. Самыми эффективными и общеупотребимыми являются искусственные нейронные сети (ИНС). Данные технологии используются в сферах безопасности и наблюдения, сканирования и создания изображений, дополненной реальности и поиска изображений. В данный момент в процессе написания находится диссертационная работа на тему «Локализация и трекинг неригидных объектов с помощью машинного обучения и методов искусственного интеллекта», цель которой заключается в анализе методов и технологий обнаружения и трекинга неригидных объектов (без фиксированной формы) и в разработке новых и улучшенных методов и технологий, способствующих робастной локализации и устойчивому трекингу таких объектов.

В настоящее время разрабатываются и успешно применяются нейронные сети, способные автоматически распознавать символы, а также находить и идентифицировать объекты. Эти технологии значительно упрощают труд человека, повышая точность и надёжность рабочих процессов путём исключения человеческого элемента. К наиболее известным архитектурам можно отнести YOLO, DAFNe, ARS-DETR, RetinaNet, Faster R-CNN, SSD.

Среди разнообразия предлагаемых решений стоит выделить технологии, относящиеся к свободному программному обеспечению. Поскольку доступность и наглядность разрабатываемого продукта способствует повышению его качества в итоговом виде. Также стоит отметить, что повышение скорости развития технологий выросло благодаря сервисам, поддерживающим концепцию свободного программного обеспечения. Пользователи могут тестировать и корректировать текущую версию программного обеспечения (ПО) путём выявления тех или иных недочётов, через двустороннюю связь с командой разработчиков через веб-сервисы для хостинга ИТ-проектов и их совместной разработки (GitHub, GitLab, GitFlic, Gitee, Gogs, RhodeCode, Launchpad и т.п.). На данный момент существует значительное количество как отдельных реализаций нейронных сетей, так и нейросетевых комплексов, отвечающих критериям классификации как свободное программное обеспечение.

Исходя из вышесказанного, цель данного доклада можно сформулировать как проведение обзора проектов открытого программного обеспечения для решения задач обнаружения и трекинга объектов с

использованием нейронных сетей и анализа эффективности качества работы разработчиков в ситуации возникновения обратной связи от пользователей.

Приоритетные архитектуры ИНС

YOLO — одна из самых популярных архитектур нейронных сетей и алгоритмов обнаружения объектов. Алгоритм YOLO разбивает входное изображение на сетку и предсказывает ограничивающие рамки и вероятности классов для каждой ячейки сетки. Он предсказывает вероятности классов и местоположения нескольких объектов за один проход по сети, что делает его более быстрым и эффективным, чем другие алгоритмы обнаружения объектов. На текущий момент существует 10 версий Yolo. Далее представлен краткий обзор самых актуальных моделей данной архитектуры [1], версий 6–10.

В модели YOLOv6 добавлен модуль ViC, который улучшает локализацию и производительность. Стратегия обучения с помощью якоря (AAT): использует преимущества парадигмы с якорями и без якорей для повышения эффективности выводов. Стратегия самодистилляции: усиливает вспомогательную ветвь регрессии во время обучения и удаляет её при выводе, чтобы избежать снижения скорости.

YOLOv7 оптимизирует процесс обучения, предлагая перепараметризованную модель и метод динамического присвоения меток, используются методы расширенного и сложного масштабирования для детектора объектов в реальном времени, позволяющие эффективно использовать параметры и вычисления. Этот метод сокращает параметры и вычисления современного детектора объектов, повышая скорость вывода и точность обнаружения.

YOLOv8 — одна из последних версий детектора объектов в реальном времени YOLO, обеспечивающая высокую точность и скорость. Сплит-головка YOLOv8 повышает точность обнаружения.

Из самых новых решений можно отметить YOLOv9 и YOLOv10. Версия YOLOv9 содержит в себе такие новаторские технологии, как программируемая градиентная информация (PGI) и обобщённая эффективная сеть агрегирования слоёв (GELAN). Модель демонстрирует значительные улучшения в эффективности, точности и адаптивности, устанавливая новые стандарты на наборе данных MS COCO. Инновации YOLOv9 глубоко укоренились в решении проблем, связанных с потерей информации в глубоких нейронных сетях. Принцип информационно-

го узкого места и инновационное использование обратимых функций занимают центральное место в его конструкции, обеспечивая высокую эффективность и точность.

YOLOv10 использует согласованное двойное назначение для обучения YOLO без NMS, что обеспечивает конкурентоспособную производительность и низкое время вывода одновременно. Эксперименты показывают, что YOLOv10-S работает в 1.8 раза быстрее чем RT-DETR-R18 при схожем AP на COCO, при этом имеет в 2.8 раза меньшее количество параметров и FLOPs. По сравнению с YOLOv9-C, YOLOv10-B имеет на 46% меньшее время задержки и на 25% меньше параметров для той же производительности.

DAFNe [2] — одноступенчатая безъякорная глубокая сеть для ориентированного обнаружения объектов. Как одноэтапная модель, она выполняет прогнозирование границ на плотной сетке по входному изображению, что делает её архитектурно более простой, а также легче оптимизируемой по сравнению с двухэтапными аналогами. Кроме того, как модель без якорей, она снижает сложность предсказания, отказываясь от использования якорей для ограничительных рамок. В DAFNe введено обобщение функции *center-ness* с учётом ориентации для произвольно ориентированных ограничительных блоков, чтобы снизить вес некачественных предсказаний, а также стратегию предсказания от центра к углу ограничительного блока, которая улучшает производительность локализации объектов. Эксперименты показывают, что DAFNe превосходит все предыдущие одноэтапные безъякорные модели на DOTA 1.0, DOTA 1.5 и UCAS-AOD и находится на одном уровне с лучшими.

ARS-DETR [3] — чувствительный к соотношению сторон детектор ориентированных объектов с трансформатором. В частности, предлагается новый метод классификации углов, вызывающий сглаживающую метку круга с учётом соотношения сторон (AR-CSL), чтобы сгладить метку угла более разумным способом и отказаться от гиперпараметра, введённого в предыдущей работе (например, CSL). Разработан вращаемый деформируемый модуль attention для поворота точек выборки под соответствующими углами и устранения несоосности между объектами региона и точками выборки. Для расчёта потери угла используется динамический весовой коэффициент в соответствии с соотношением сторон.

RetinaNet [4] — одноступенчатая сеть обнаружения объектов. Основное внимание уделяется устранению дисбаланса между обучаю-

щими выборками переднего плана и фона, который является важнейшей причиной низкого качества обнаружения многих других одноступенчатых сетей. Предлагается фокальная потеря для замены широко принятой перекрёстной энтропийной потери для задачи классификации. Используя потерю фокуса, *RetinaNe* может распределять больше весов для плохо классифицированных жёстких выборок во время обучения и улучшать обобщение обученной сети для тестирования данных.

Faster R-CNN [5] — платформа обнаружения объектов, основанная на глубоких свёрточных сетях, которая включает в себя сеть предложений регионов (RPN) и сеть обнаружения объектов. Обе сети обучены совместно использовать свёрточные слои для быстрого тестирования.

SSD.Pytorch [6] — библиотека, широко используемая в телекоммуникациях, средствах массовой информации, развлечениях, искусственном интеллекте, компьютерном зрении, глубоком обучении, *Pytorch*, приложениях.

Использование ИНС в проектах открытого программного обеспечения по обнаружению объектов

Концепция свободного программного обеспечения включает в себя разработки для решения задач в самых разных областях жизни. В современном мире с ростом потребностей в обработке значительных объёмов данных, зачастую в режиме реального времени, возникает необходимость в специализированных программных решениях. Технологии свободного программного обеспечения для обнаружения объектов имеют множество преимуществ, а именно — точность и скорость распознавания, человеческий мозг не может конкурировать с данными показателям. Так же к важным плюсам можно отнести сокращение затрат на реализацию процесса обнаружения объектов, за счёт сокращения времени, которое потребовалось бы человеку для выполнения данной задачи. Отсутствие предвзятости при обработке данных. В качестве примера можно рассмотреть следующие проекты.

Ultralytics [1] — набор нейросетевых архитектур, каждая из которых адаптирована к конкретным задачам. Их широкий спектр включает такие задачи как обнаружение и классификация объектов, сегментация изображений, оценка позы и отслеживание нескольких объектов.

AI Basketball Analysis [7] — веб-приложение и API на базе искусственного интеллекта (ИИ), которое анализирует баскетбольные броски и позы игроков в момент броска, построенные на основе концепции обнаружения объектов. В этом проекте есть три основные функции: анализ снимков, обнаружение снимков и API обнаружения. Проект обнаружения объектов реализован на Python с использованием библиотеки с открытым исходным кодом OpenPose. Проект разработан с использованием концепции трансферного обучения, для обучения используется более быстрая модель — RCNN, предварительно обученная на наборе данных COCO.

AVOD [8] — проект разработан с целью 3D-обнаружения объектов для автономных самоуправляемых автомобилей. При реализации использовались Python, OpenCV и Tensorflow. В качестве набора данных для обучения использовался Kitti Object Detection Dataset, результаты сравнивались с различными другими опубликованными методами в тестах Kitti 3D object и BCV Benchmarks. Набор данных Kitti включает изображения восьми различных классов: легковой автомобиль, фургон, грузовик, пешеход, сидящий человек, велосипедист, трамвай, Разное и DontCare.

Vehicle Counting [9] — проект с открытым исходным кодом, в центре которого обнаружение, отслеживание и подсчёт транспортных средств. Этот проект обнаружения объектов также предоставляет прогнозы скорости, цвета, размера и направления движения транспортного средства в режиме реального времени с использованием TensorFlow Object Detection API. При реализации этого проекта используются TensorFlow, OpenCV и Python, а модель, используемая для обнаружения транспортных средств, — SSD.Pytorch с mobilenet. В настоящее время в этом проекте реализована классификация пяти транспортных средств: автобус, легковой автомобиль, велосипед, грузовик и мотоцикл.

Open-mmlab [10] — система алгоритмов компьютерного зрения с открытым исходным кодом. Некоторые, основные, проекты в OpenMMLab: MMEEngine, MMCV, MMPreTrain, MMDetection, MMDetection3D, MMRotate, MMYOLO, MMSegmentation, MMTracking. MMEEngine — базовая библиотека для обучения моделей глубокого обучения, она служит механизмом обучения всех кодовых баз OpenMMLab которые поддерживают сотни алгоритмов в различных областях исследований. Более того, MMEEngine также является универсальным инструментом для применения к проектам, отличным от OpenMMLab. MMCV — библиотека

для исследований компьютерного зрения, предоставляющая следующие функциональные возможности: обработка изображений/ видео, визуализация изображений и аннотаций, преобразование изображений, различные архитектуры CNN, высококачественная реализация общего процессора и операций CUDA. Немаловажной особенностью является возможность использования на различных операционных системах, в частности Linux, Windows и macOS. **MMPreTrain** — набор инструментов для предварительной подготовки. **MMDetection** — набор инструментов обнаружения объектов. **MMDetection3D** — набор инструментов для обнаружения 3D-объектов. **MMPRotate** — набор инструментов для работы с повернутыми объектами. **MMYOLO** — инструментарий для алгоритмов серии YOLO на основе PyTorch и MMDetection. **MMSegmentation** — набор инструментов семантической сегментации. **MMTracking** — набор инструментов для трекинга объектов по видеоданным.

Torchvision [11] содержит определения моделей для решения различных задач, включая: классификацию изображений, семантическую сегментацию по пикселям, обнаружение объектов, сегментацию экземпляров, обнаружение ключевых точек человека, классификацию видео и оптический поток.

Анализ репозитория

Одним из ключевых преимуществ концепции свободного программного обеспечения является возможность коллективного совершенствования программного обеспечения путём взаимодействия разработчиков и пользователей через специализированные сервисы. Пользователи могут свободно тестировать программное обеспечение и вносить свои комментарии по корректности работы и предлагать свои дополнения. Разработчики, в свою очередь, могут повысить качество продукта при подобном взаимодействии.

Анализируя качество работы репозитория следует выделить следующие критерии: тематика репозитория, опыт использования пользователями, впечатление о качестве поддержки команды разработчиков, частота обновления, частота изменения API на несовместимый, наличие документации, наличие примеров кода, кроссплатформенность и т. п. Анализ проводится на основе полученных данных из репозитория специализированных web-сервисов.

Для более наглядного сравнения составлена таблица 1

Таблица 1: Сравнение репозиторий.

Репозиторий	Число звёзд	Качество поддержки	Частота обновления	Частота изменения API	Документация	Примеры кода	Кросс-платформенность
Ultralytics	24300	Разработчики активно реагируют на возникающие проблемы	Высокая	Низкая	Да	Да	Да
AI Basketball Analysis	935	Команда плохо реагирует на возникающие проблемы	Средняя	Средняя	Да	Да	Данных нет
AVOD	920	Команда плохо реагирует на возникающие проблемы	Низкая	Низкая	Да	Да	Данных нет
Vehicle Counting	0	Команда плохо реагирует на возникающие проблемы	Низкая	Низкая	Да	Да	Данных нет
Open-mmlab	112000	Разработчики активно реагируют на возникающие проблемы	Высокая	Высокая	Да	Да	Да
Torchvision	15600	Разработчики активно реагируют на возникающие проблемы	Высокая	Высокая	Да	Да	Да

Заключение

Концепция свободного программного обеспечения благотворно сказывается на качестве и скорости разработки. По результатам проведённого анализа наблюдается взаимосвязь между качеством работы команды разработчиков с пользователями и общим качеством продукта. Принципы свободного ПО способствуют наглядности разработки и совершенствованию программного продукта в процессе эксплуатации и тестирования неограниченным количеством пользователей. Также стоит отметить, что использование специализированных веб-сервисов помогает определить уровень востребованности производимой технологии, возможность сравнивать реализацию с прочими подобными технологиями и при необходимости корректировать версии.

Исследования выполнены за счёт гранта Российского научного фонда № 22-11-20001 (см. <https://rscf.ru/project/22-11-20001/>) и гранта в форме субсидии из областного бюджета организациям Ярославской области.

Литература

- [1] Models Supported by Ultralytics, 2024. <https://github.com/ultralytics/ultralytics/blob/main/docs/en/models/index.md>
- [2] DAFNe: A One-Stage Anchor-Free Approach for Oriented Object Detection. Github, 2022. <https://github.com/braun-steven/DAFNe>
- [3] ARS-DETR: Aspect Ratio Sensitive Oriented Object Detection with Transformer. Github, 2023. <https://github.com/httle/ARS-DETR>
- [4] pytorch-retinanet. Github, 2021. <https://github.com/yhenon/pytorch-retinanet>
- [5] Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. Github, 2018. https://github.com/ShaoqingRen/faster_rcnn
- [6] SSD: Single Shot MultiBox Object Detector, in PyTorch. Github, 2019. <https://github.com/amdegroot/ssd.pytorch>
- [7] AI Basketball Analysis. Github, 2022. <https://github.com/chonyy/AI-basketball-analysis>
- [8] Aggregate View Object Detection. Github, 2018. <https://github.com/kujason/avod>
- [9] Video-based Vehicle Counting System. Github, 2019. <https://github.com/pubTools/Vehicle-Counting>
- [10] Open-mmlab. Github, 2024. <https://github.com/open-mmlab>
- [11] torchvision. Github, 2024. <https://github.com/pytorch/vision>

Александр Нозик
Москва, МФТИ

Проект: KMath и другие проекты <https://github.com/SciProgCentre>

Научная экосистема KScience

Аннотация

Наука — это передний край человеческого знания. Самые талантливые люди, самое современно оборудование, самые интересные задачи, но.., к сожалению, часто далеко не самое современное программное обеспечение. История старая и в основном связана с тем, что в научные метрики до сих пор не включают разработки ПО. Но сейчас появились попытки исправить эту досадную ситуацию. KScience — это одна из таких попыток. Исходно появившаяся в сфере физике частиц, сейчас экосистема используется в том числе для промышленных задач.

Ключевые слова: *научное ПО, Kotlin, программная экосистема, физика частиц, экосистема открытого ПО.*

В этом докладе мы обсудим, какие предпосылки есть для создания экосистемы открытого ПО, какие проблемы есть на этом пути, а также то, каким образом открытая экосистема может сама себя финансировать.

Хорошо известно, что современное программирование во многом появилось в научной среде и во многом в физике частиц. Например, протокол HTTP, на котором построен современный Интернет, был придуман в 1990 году сэром Тимоти Бернесом-Ли и соавторами для работы с данными физических экспериментов. Многие разработки, связанные с языками программирования, также связаны с фундаментальной физикой. Тем не менее, со временем программная индустрия шагнула далеко вперёд, а программирование в науке продвинулось довольно слабо. Таким образом сформировался «раскол» между уровнем технологий и инструментария в промышленном программировании и в научном программировании. На этом фоне возникла потребность в разработке новых технологий и подходов в программировании специально для научного применения.

KScience — это экосистема открытых библиотек, распространяемых под лицензией Apache 2.0, написанных при помощи технологии Kotlin-multiplatform. Многие из этих проектов возникли в рамках работ по сбору и обработке данных в эксперименте Troitsk nu-mass [1, 2]. Несмотря на небольшой по меркам физики частиц размер

установки, там необходим очень сложный многоступенчатый анализ данных с большой вариативностью входных параметров и возможностью сравнения результатов. Первые версии оптимизатора для обработки данных для этого эксперимента были созданы в 2005 году Фёдором Ткачёвым на языке **Component Pascal** (платформа **BlackBox**) и основывались на разработанном им методе квази-оптимальных вёсов [3].

В процессе разработки стало понятно, что одного алгоритма многомерной оптимизации недостаточно для того, чтобы закрыть все потребности обработки эксперимента, а нужную гибкость невозможно реализовать на «учебном» языке **Component Pascal**. В результате код был переписан на язык **Java**, и была создана программная платформа для автоматизированного анализа данных, позднее получившая название **DataForge**. В 2011–2013 годах при помощи этой платформы были обработаны данные эксперимента (как по поиску массы электронного нейтрино, так и по поиску примеси стерильного нейтрино) и получены лучшие в мире на тот момент результаты.

В 2016 году была начата плавная миграция инструментов на язык **Kotlin**, который обладает полной обратной совместимостью с **Java** и при этом является более гибким и компактным, что делает его более привлекательным с точки зрения научного программирования. В 2017–2018 годах по мере плавной стабилизации технологии **Kotlin-multiplatform**, позволяющей компилировать язык на **Kotlin** под разные платформы исполнения (**JVM**, **JS**, **LLVM**/нативный машинный код, а теперь ещё и **Wasm**), платформа была полностью переписана на этой технологии с учётом новых разработок в дизайне программных интерфейсов.

В настоящее время экосистема библиотек **KScience** включает следующие библиотеки:

- **KMath**¹ — самая крупная математическая библиотека на языке **Kotlin**. **KMath** использует уникальный подход к организации математических операций, где реализация операций не является ответственностью элементов, над которыми производятся операции (например чисел), а является частью контекста — алгебры. Таким образом, над одним и тем же элементом может существовать несколько разных алгебр.

¹<https://github.com/SciProgCentre/kmath>

- **DataForge-core**² — основа системы управления метаданными и задачами (по принципу «процессора метаданных»).
- **VisionForge**³ — набор библиотек распределённой визуализации. Система исходно создавалась в основном для реализации систем визуализации событий на ускорителе NICA (трёхмерная графика), но сейчас используется для значительно более широкого круга задач.
- **Controls-kt**⁴ — платформа нового поколения для асинхронной работы с приборами. Включает инструменты для создания легковесной системы управления сбором данных, управления оборудованием, моделирования и создания аналитических сервисов.
- **Maps-kt**⁵ — набор библиотек для картографии и динамической картографической визуализации.

Кроме того, существует ряд библиотек, разработка которых находится на стадии прототипирования.

В настоящее время при помощи инструментов из экосистемы KScience создаются не только научные приложения, но и ведётся разработка прикладных (коммерческих) проектов. Это позволяет поддерживать экосистему в актуальном состоянии и делать новые разработки.

Литература

- [1] Aseev V. N. et al., An upper limit on electron antineutrino mass from Troitsk experiment Phys. Rev. D84 112003 <https://journals.aps.org/prd/abstract/10.1103/PhysRevD.84.112003>.
- [2] Alexander Nozik, Declarative analysis in ‘Troitsk nu-mass’ experiment, <https://iopscience.iop.org/article/10.1088/1742-6596/1525/1/012024>.
- [3] Fyodor Tkachov, Quasi-optimal weights: a versatile tool of data analysis, <https://doi.org/10.1088/1742-6596/523/1/012030>.

²<https://github.com/SciProgCentre/dataforge-core>

³<https://github.com/SciProgCentre/visionforge>

⁴<https://github.com/SciProgCentre/controls-kt>

⁵<https://github.com/SciProgCentre/maps-kt>

Владислав Андреев

Архангельск, «Северный (Арктический) федеральный университет имени М. В. Ломоносова»

Использование свободного ПО в учебном процессе по специальности «DevOps-инженер»

Аннотация

В докладе представлены примеры свободного программного обеспечения, которые активно используются в сфере образования для специальности «DevOps-инженер». Какие задачи можно решить благодаря свободному ПО. Рассказывается о причинах перехода.

Ключевые слова: *свободное программное обеспечение, DevOps, Linux, информационные технологии, образование.*

Процесс перехода на свободное ПО идёт по всему миру, с каждым годом набирая всё более массовый характер. По результатам опроса, проводимого среди 87 тысяч респондентов на StackOverflow [1], около 50% использовали различные дистрибутивы Linux как основную ОС. В прошлом году данная цифра составляла около 40%, а два года назад — чуть более 23%. Это свидетельствует о том, что пользователи всё больше склонны выбирать свободное программное обеспечение как основные инструменты для работы и повседневного использования.

В рамках политики импортозамещения российские компании стали переходить либо на отечественное программное обеспечение, либо на свободное ПО. В первую очередь, это относится к государственным учреждениям, таким как университеты, школы, правоохранительные органы, банки и т. д. Данный переход позволяет закрывать все необходимые потребности в программном обеспечении, несмотря на уход зарубежных компаний.

Данная смена ПО хорошо видна в учебном процессе, особенно в высших учебных заведениях. При работе со свободным программным обеспечением студенты получают необходимые знания и практические навыки для будущей работы, так как большинство организаций работают с теми же программами.

Например, студенты, которые обучаются на направлениях, связанных с информационными технологиями, нередко прибегают к различным языкам программирования, используя для работы с ними редакторы кода и среды разработки. Создают виртуальные машины, устанавливая там дистрибутивы GNU/Linux. Студенты активно работают

с контейнерами. Настраивают непрерывную интеграцию, редактируют под себя конфигурационные файлы веб-серверов, производят мониторинг сети и работают с облаками, учатся управлять системами контроля версиями, используют приложения для изучения и преподавания математики.

Как можно заметить, свободное ПО очень востребовано в университетских кругах. Множество различных программ выполняет огромное количество функций, что позволяет студентам развиваться в их сфере. Остановимся поподробнее на ряде программ и разберёмся, почему они так востребованы и каким образом были внедрены в учебный процесс.

Начнём с самого востребованного программного обеспечения у всех пользователей, офисного пакета. Без данного ПО невозможно представить ни одну работу в университете. Курсовые проекты, различные отчёты, составление графиков, таблиц и презентаций — всё это делается в программах, которые входят в офисные пакеты. Самыми популярными на данный момент являются пакеты LibreOffice [2] и OpenOffice [3].

Это кроссплатформенные офисные пакеты с открытым исходным кодом, которые распространяются по лицензиям MPL 2.0 и LGPL соответственно.

Студенты, которые осваивают информационные технологии, активно пользуются свободным ПО для широкого круга задач. Самое банальное — это использование языков программирования для написания своих приложений, сервисов, игр и программ. Самыми популярными на данный момент являются языки: Python, C++, Kotlin и SQL. Для работы с ними чаще всего используют универсальный текстовый редактор — Visual Studio Code [4]. Он обладает рядом полезных функций и возможностей, которые выделяют его на фоне аналогов.

Для виртуализации используется Oracle VM VirtualBox [5]. С помощью данной программы можно запускать одну операционную систему внутри другой, что является очень удобным инструментом, когда нужно работать с несколькими ОС одновременно.

Чтобы сохранять информацию и в дальнейшем с ней работать, нужно использовать базу данных. Самым распространённым вариантом с открытым исходным кодом является PostgreSQL [6], реляционная система управления базами данных (СУБД), которая реализована для множества ОС.

Для обучения непрерывной интеграции программного обеспечения используются `Gitlab` [7] и `Jenkins` [8]. С помощью данных инструментов студенты знакомятся с методологией `DevOps`. `DevOps` — это методология автоматизации технологических процессов сборки, настройки и развёртывания программного обеспечения.

Работа с контейнеризацией осуществляется с помощью `Podman` [9] в связке с `Kubernetes` [10]. Главным преимуществом `Podman` является отсутствие дополнительного сервиса для работы. Благодаря `Kubernetes` можно заниматься оркестрацией данных контейнеров: координировать, размещать, развёртывать и масштабировать.

`Apache` [11] и `NGINX` [12] — кроссплатформенные веб-сервера с открытым исходным кодом, с помощью которых можно устанавливать соединение между пользователем и сервером.

Для обучения мониторинга сети используют `Zabbix` [13] — свободная система мониторинга статусов разнообразных сервисов компьютерной сети, серверов и сетевого оборудования. Данная система может сохранять данные в базу данных `PostgreSQL`, о которой говорилось выше.

Работа с облачной инфраструктурой осуществляется путём работы с платформой `OpenNebula` [14]. Основными задачами `OpenNebula` являются виртуализация центров обработки данных и облачные развёртывания.

Студенты активно используют распределённую систему управления версиями `Git` [15]. За счёт неё у разработчика ПО есть возможность переключаться между версиями, следить за историей разработки и работать в команде над одним проектом.

Всё вышеперечисленное программное обеспечение, на котором работают студенты, помимо основных занятий изучаются на курсах дополнительного образования [16]. Благодаря им студенты получают не только дополнительный опыт, но и сертификат о прохождении курса или удостоверение о повышении квалификации, что впоследствии поможет им устроиться на работу.

Как можно заметить, огромное количество задач решается с помощью свободного программного обеспечения и с каждым годом свободного ПО становится всё больше, что положительно сказывается на разработке программных решений.

Литература

- [1] <https://survey.stackoverflow.co/2023/#section-most-popular-technologies-operating-system>
- [2] <https://ru.libreoffice.org/>
- [3] <https://www.openoffice.org/ru/>
- [4] <https://code.visualstudio.com/>
- [5] <https://www.virtualbox.org/>
- [6] <https://www.postgresql.org/>
- [7] <https://gitlab.com/>
- [8] <https://www.jenkins.io/>
- [9] <https://podman.io/>
- [10] <https://kubernetes.io/>
- [11] <https://www.apache.org/>
- [12] <https://nginx.org/ru/>
- [13] <https://www.zabbix.com/ru>
- [14] <https://opennebula.io/>
- [15] <https://git-scm.com/>
- [16] <https://narfu.ru/hsitas/obuchenie/dopolnitelnoe-obrazovanie/>

Олег Неманов

Кремёнки, МОУ СОШ им. генерала Захаркина И. Г.

Проект: libsg <https://github.com/lego12239/libsg>

Наш опыт использования СПО на уроках информатики

Аннотация
Так получилось, что возникла необходимость, помимо своей основной работы программистом, проводить уроки информатики в старших классах школы. После исследования вопроса была выбрана программа для вузов и школ Дединского И. Р.

Была произведена работа по переводу класса информатики на Linux-based ОС, подобрано необходимое ПО и, так как оригинальная графическая библиотека из учебной программы Дединского И. Р. написана под Windows, была написана библиотека-обёртка для SDL, что бы использовать её для обучения.

Ключевые слова: *программирование, СПО, Си, дети, цветы жизни.*

Основные тезисы доклада:

- выбор в качестве программы для преподавания программирования в старших классах средней школы программу Дединского И. Р.;
- обучение программированию с нуля на Си в качестве первого ЯП;
- процесс перевода класса информатики на СПО (что хотели и что в итоге получилось), выбор и установка ОС, рхе-загрузка;
- проблема выполнения домашних заданий (дома, в основном, Windows); подготовка образов VM с Linux-based ОС и доступ по rdp на школьную машину с учётными записями учеников;
- создание страницы в Интернет с необходимой справочной информацией для ученика;
- создание библиотеки обёртки для SDL — libsg. Библиотека скрывает нюансы работы SDL, которые не нужны обучающемуся на первых порах и могут его запутать;
- другое ПО используемое в процессе обучения — редактор, компилятор, отладчик, среда разработки;
- что хотелось бы доработать и изменить, направление дальнейшей работы.

Литература

- [1] Дединский И. Р., Презентация по методике преподавания — <http://storage.ded32.net.ru/Lib/Doc/AnalyticApproach2010.pdf>
- [2] Дединский И. Р., Учебные программы — <http://storage.ded32.net.ru/Lib/Doc/Curriculum-5-11.pdf>
- [3] Проект libsg — <https://github.com/lego12239/libsg>
- [4] Справочная страница ученик — <http://c.kremenki.su/>

Сергей Скрипов
Челябинск, ЧелГУ

К вопросу об использовании свободного программного обеспечения для управления виртуальными машинами в учебном процессе

Аннотация

Для организации учебного процесса для направлений подготовки в области информационных технологий необходимо иметь возможность массово создавать виртуальные машины, а также управлять ими. В работе рассматривается свободное программное обеспечение для запуска и управления виртуальными машинами, возможные сценарии внедрения виртуальных машин в учебный процесс, а также возникающие при этом проблемы.

Ключевые слова: *виртуальная машина, libvirt, OpenNebula.*

Для проведения лабораторных и практических занятий по дисциплинам, связанным с информационными технологиями, требуется использование виртуальных машин. Это связано с тем, что значительное количество программного обеспечения использовать на реальных компьютерах неудобно и небезопасно. Например, для дисциплины «Защита информации» требуется предоставить студентам заведомо уязвимые сервисы, для которых нужно воспроизвести атаку. Поэтому был развёрнут кластер, позволяющий быстро создавать виртуальные машины, доступные по сети, а также управлять ими [1, 2]. Для организации кластера был использован следующий стек:

- На физических серверах установлена операционная система CentOS 7. Также для выбранного стека хорошо подойдут актуальные LTS версии Ubuntu и AlmaLinux.
- В качестве гипервизора используется `qemu-kvm`.
- Для управления виртуальными машинами используется `libvirt` [4].
- Для более удобного управления используется `OpenNebula` [5]. Этот продукт позволяет автоматизировать и упростить процесс управления виртуальными машинами.

В процессе эксплуатации были выявлены реально применяемые сценарии использования виртуальных машин в учебном процессе:

- На виртуальных машинах разворачиваются службы инфраструктуры, например общие папки, VPN.
- На виртуальной машине настраивается отдельная служба, необходимая для учебного процесса. К самой виртуальной машине студенты доступа не имеют.
- На виртуальной машине для каждого студента создаётся неприВИлегированная учётная запись, студент работает через SSH.
- Для каждого студента создаётся одна или несколько виртуальных машин, студенты имеют на своих виртуальных машинах полные привилегии.
- Для проведения лабораторных работ по сетевым технологиям [2] студенты используют отдельный сервер, на котором у них имеется доступ напрямую к `OpenNebula`.

В отличие от коммерческих продуктов, которые работают «из коробки», свободное программное обеспечение требует доработки:

- Для работы с большими файлами виртуальных дисков на физическом сервере необходимо настроить параметры ядра `vm.dirty_background_bytes` и `vm.dirty_bytes` [3].
- Для эффективного использования памяти на физическом сервере необходимо создать большой раздел подкачки.
- `OpenNebula` учитывает расход памяти и ресурсов процессора даже для остановленных виртуальных машин. Для адекватной проверки остатка этих ресурсов в некоторые файлы `OpenNebula` были внесены изменения.

- Библиотека `libvirt` имеет средства, позволяющие корректно останавливать и запускать виртуальные машины при перезагрузке сервера. Однако `OpenNebula` не создаёт полноценный `libvirt` домен, поэтому такие возможности использовать нельзя. При остановке демона `OpenNebula` виртуальные машины продолжают работать и будут некорректно выключены в процессе перезагрузки. При старте демона виртуальные машины автоматически не запускаются. Поэтому необходимо написание сценария, обеспечивающего корректную остановку и запуск виртуальных машин.
- Для создания собственного образа диска виртуальных машин необходимо создать эталонную виртуальную машину, произвести необходимые настройки и установку программного обеспечения. После этого необходимо найти файл виртуального жёсткого диска и загрузить его как новый образ в `OpenNebula`. После этого получившийся образ можно использовать в шаблонах.

Таким образом, после доработки был развёрнут кластер, который в настоящее время активно применяется в учебном процессе.

Литература

- [1] С. А. Скрипов Реализация многопользовательской виртуальной лаборатории на основе облачных технологий // Информационные технологии и системы: Труды Шестой Международной научной конференции. — Издательство Челябинского государственного университета, 2017. — С. 275–276
- [2] С. А. Скрипов Реализация стенда на основе облачных технологий для проведения лабораторных работ по дисциплинам, связанным с компьютерными сетями // Инновационные, информационные и коммуникационные технологии: Сборник трудов XIX Международной научно-практической конференции. — Ассоциация выпускников и сотрудников ВВИА им. проф. Жуковского, 2022. — С. 27–30
- [3] Documentation for `/proc/sys/vm/` [Электронный ресурс] / Электрон. дан. — Режим доступа: <https://docs.kernel.org/admin-guide/sysctl/vm.html>, свободный.
- [4] Libvirt Documentation [Электронный ресурс] / Электрон. дан. — Режим доступа: <https://libvirt.org/docs.html>, свободный.
- [5] OpenNebula 6.8 Documentation [Электронный ресурс] / Электрон. дан. — Режим доступа: <https://docs.opennebula.io/6.8>, свободный.

Андрей Галактионов, Лев Багнюк, Егор Погорелко
Москва, Кафедра ИУ10 «Защита информации» МГТУ им. Н. Э. Баумана
Проект: Hashpass <https://github.com/Bunnyton/hashpass>

Подходы к созданию комплекса программных средств автоматизированного проведения и проверки учебных лабораторных работ

Аннотация

В докладе будет представлен подход к созданию комплекса программных средств автоматизированного проведения и проверки учебных лабораторных работ. Ключевыми особенностями подхода являются автоматизированное создание персонализированных изолированных окружений на основе виртуальных машин и средств изоляции процессов Linux, а также упрощённое описание заданий и критериев их успешного выполнения.

Ключевые слова: *Linux, интерактивное обучение, современный подход.*

В рамках данной работы будет рассказано о подходе к разработке комплекса программных средств, который позволит проводить интерактивные лабораторные работы. Целью создания данного ПО является улучшение и упрощение образовательного процесса студентов.

Данный комплекс разрабатывается, в первую очередь, для проведения лабораторных работ по изучению Linux. Опытным путём обнаружено, что для достижения определённой задачи, будь то создание файла или смена пароля, может использоваться огромное количество различных команд и их комбинаций. Для того, чтобы достигать максимальной эффективности в данном ПО, используется подход, анализирующий не конкретные команды, а результат их выполнения. Конкретно, анализируется изменение файловой системы, а также вывод после выполнения команды.

Можно подумать, что можно ошибиться в командах или ввести их с избытком, например, постоянно используя команду `ls`. Данная проблема решена использованием принципа «разделяй и властвуй», то есть общее количество команд разделено на этапы, которые достигают определённых целей. Таким образом, можно выделить точки, где комплекс программных средств собирает данные.

Для максимальной реалистичности в данном комплексе используется `systemd-nspawn`, что позволяет эмулировать полноценную работу ОС. Для максимальной шаблонизации задач используется `overlay-fs`, что позволяет сэкономить пространство и облегчить опыт использования.

Таким образом, можно создать «задание», организовав пространство и зафиксировав решение. Если пользователь, выполняющий задание, осуществит действия, которые приведут его к тем же результатам, что и создателя задания, то оно будет засчитано как выполненное.

Александр Зубарев

Архангельск, Северный (Арктический) федеральный университет имени М. В. Ломоносова

Использование систем поиска уязвимостей локальных и сетевых систем

Аннотация

В данной статье рассматриваются вопросы применения свободного программного обеспечения для выявления уязвимостей в информационных системах, связанных с локальными и сетевыми индикаторами. Цель статьи — рассмотреть вопросы, связанные с применением инструментов по выявлению уязвимостей в дальнейшей работе. В статье приведены популярные библиотеки и методики сканирования.

Ключевые слова: *уязвимость, тестирования на проникновение, информационная безопасность.*

Применение лицензионных и сертифицированных средств защиты в образовательном процессе играет достаточно большую роль в подготовке кадров в области защиты информационных и автоматизированных систем. В целом, применение средств защиты как основа для обучения позволяет подготовить специалиста, который может применять средства защиты информации на предприятиях там, где они используются. Одна из ключевых проблем — это реализация практико-ориентированного подхода и формирование рамок, в которых специалист оказывается не способен пользоваться другим технологиями или выполнять глубокую настройку. В большинстве инструментов и программ для выполнения поиска уязвимостей локальных и сетевых систем используется свободное программное обеспечение, являющееся

в основе своей гибким инструментом, который можно применять как основу как для научно-исследовательских работ, так и для разработки научных проектов.

Как показывает практика, свободное программное обеспечение и популярные ресурсы, такие как `github`, содержат достаточно большое количество решений для обнаружения угроз и уязвимостей. При том, что большинство корпоративных продуктов, действующих сейчас на рынке для применения анализа методами `SAST` и `DAST`, были или являются свободным программным обеспечением. Это даёт возможность их анализировать, изучать применять и, как следствие, выполнять разработку, дорабатывать и разрабатывать свой пул решений.

В рамках работы по подготовке специалистов в области информационной безопасности в целом появляется проблема использования различных продуктов в процессе обучения. Большинство работодателей используют свой стек технологий на различных решениях вендора. Использование базовых и открытых технологических решений позволяет выделять принцип работы с системами и, как правило, определять узкие места. В том числе, дорабатывать по текущим потребностям. К примеру, разработка анализаторов сетевых серверов на основе `nmap` с дополнением вопроса оптимизации и добавлением характерных признаков определения нужных точек.

Использование `nmap` обусловлено тем, что на популярном языке программирования `Python` имеется не менее 210 проектов, которые позволяют автоматизировать задачи под конкретные потребности. Например, библиотека `python3-nmap` позволяет быстро получать данные для дальнейшего анализа и использования в сопутствующих проектах. Атаки, которые позволяют выявить информации, реализованы в `python3-nmap`:

- `nmap_dns_brute_script`;
- `nmap_version_detection`;
- `nmap_stealth_scan`;
- `nmap_detect_firewall`;
- `nmap_os_detection`;
- `nmap_subnet_scan`;
- `nmap_list_scan`.

Кроме этого, выявление следующих атак:

- TCP SYN Scan (-sS);
- TCP connect() scan (-sT);
- FIN Scan (-sF);
- Ping Scan (-sP);
- Idle Scan (-sI).

Получение информации после выполнения функций модуля позволяет получать стерилизованный набор данных, который можно использовать в дальнейшем.

Выполнение команды `nmap.nmap_os_detection("192.168.178.2")` возвращает следующий набор данных:

```
[
  {
    "accuracy": "100",
    "cpe": "cpe:/o:linux:linux_kernel:2.6",
    "line": "45249",
    "name": "Linux_2.6.14_2.6.34",
    "osclass": {
      "accuracy": "100",
      "osfamily": "Linux",
      "osgen": "2.6.X",
      "type": "general_purpose",
      "vendor": "Linux"
    }
  },
  {
    "accuracy": "100",
    "cpe": "cpe:/o:linux:linux_kernel:2.6.17",
    "line": "45775",
    "name": "Linux_2.6.17",
    "osclass": {
      "accuracy": "100",
      "osfamily": "Linux",
      "osgen": "2.6.X",
      "type": "general_purpose",
      "vendor": "Linux"
    }
  },
  {
    "accuracy": "100",
    "cpe": "cpe:/o:linux:linux_kernel:2.6.17",
    "line": "45811",
    "name": "Linux_2.6.17(Mandriva)",
    "osclass": {
      "accuracy": "100",
      "osfamily": "Linux",
      "osgen": "2.6.X",
      "type": "general_purpose",
      "vendor": "Linux"
    }
  },
  {
    "accuracy": "100",
    "cpe": "cpe:/o:linux:linux_kernel:3.13",
    "line": "60884",
    "name": "Linux_3.13",
    "osclass": {
```



```

        "accuracy": "100",
        "osfamily": "Linux",
        "osgen": "3.X",
        "type": "general_purpose",
        "vendor": "Linux"
    }
}
1

```

Полученные данные можно обработать и занести в базы данных для дальнейшего использования. В общей тенденции, обучающийся может использовать как готовые модули, так и разрабатывать свои, что впоследствии позволяет выявлять потребность в конкретных данных и получать их. Кроме решения **nmap** используются: **Wpscan**, **Joomscan**, **Nmap**, **Nessus**, **Nikto**, **metasploit**.

Литература

- [1] *Бирюков А. А.* Информационная безопасность: защита и нападение. — Москва: ДМК Пресс, 2017. 474 с.
- [2] Nmap Reference Guide. Documentation / NMAP.ORG. <https://nmap.org/docs.html>
- [3] Search Engine for the Internet of Everything / Shodan <https://www.shodan.io/>
- [4] Python3-nmap 1.6.0 <https://pypi.org/project/python3-nmap/>

Научное издание
ДЕВЯТНАДЦАТАЯ КОНФЕРЕНЦИЯ
«СВОБОДНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ
В ВЫСШЕЙ ШКОЛЕ»

Сборник материалов конференции

Переславль-Залесский
28–30 июня 2024 года

Оформление обложки: А. В. Надточенко.
Вёрстка: В. Л. Чёрный.
Редактура: В. Л. Чёрный.

Отпечатано с готового оригинал-макета

Подписано в печать 18.06.2024

Формат 60х90 1/16 Усл.пч.л. 12.25

Тираж 250 экз. Изд. 90

Издательство ООО «МАКС Пресс»

Лицензия ИД N 00510 от 01.12.99 г.

119992, ГСП-2, Москва, Ленинские горы, МГУ им. М. В. Ломоносова,
2-й учебный корпус, 527 к

Тел. 8(495)939-3890/91. Тел./Факс 8(495)939-3893

Отпечатано в полном соответствии с качеством
предоставленных материалов в ООО «Фотоэксперт»
109316, г. Москва, Волгоградский проспект, д. 42,
корп. 5, эт. 1, пом. I, ком. 6.3-23Н